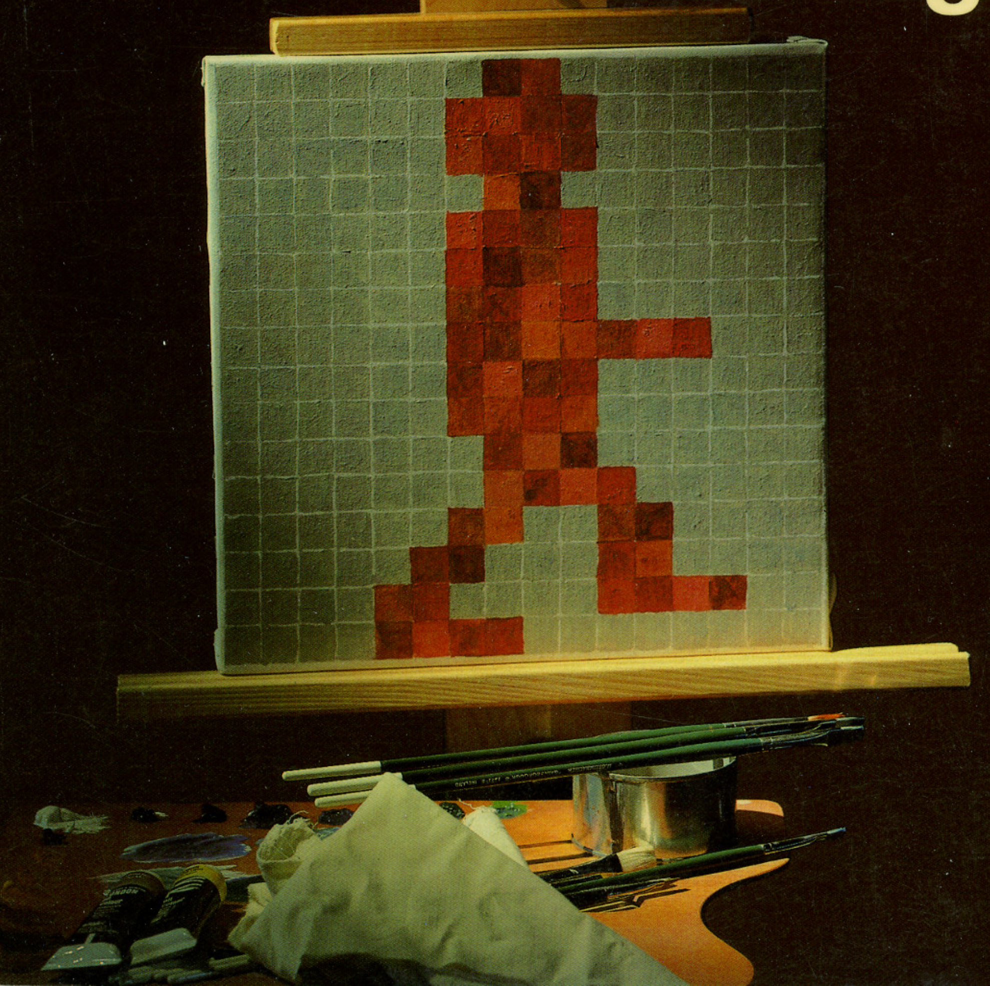
micro COMPUTER books

David D. Thornburg

Computer Art and Animation

A User's Guide to TI-99/4A Color Logo



Computer Art and Animation

A User's Guide to TI-99/4A Color Logo

David D. Thornburg

ADDISON-WESLEY PUBLISHING COMPANY

Reading, Massachusetts • Menlo Park, California
London • Amsterdam • Don Mills, Ontario • Sydney

This book is in the **Addison-Wesley Microcomputer Books** Popular Series

Cover Design—*Marshall Henrichs*
Design—*Ellen Klempner*

Library of Congress Cataloging in Publication Data
Thornburg, David D.
Computer art and animation.

1. TI 99/4A (Computer)—Programming. 2. LOGO (Computer program language) 3. Computer graphics.
I. Title.
QA76.8.T133T53 1984 001.64'43 84-12305
ISBN 0-201-07958-5 (pbk.)

ISBN 0-201-07958-5
ABCDEFGHIJ-HA-89876543

Preface

This book about computer graphics was written for two types of people—artists who are interested in computers and computer enthusiasts who are interested in art. It is possible that you fall in neither of these categories, in which case this book may stimulate your interest in both areas.

Many artists feel that computers are outside their range of comprehension, and many computerists feel the same way about art. If you fit in either of these groupings, perhaps the rest of this preface will help you toward another viewpoint.

Artists Anxious about Computers

I've met many very creative artists who freeze up when they hear the word *computer*. The idea that a faceless, emotionless symbol-manipulation tool could be used for creative expression is met with disbelief. Of course, some people think that hammers and chisels are pretty cold and faceless too, but that doesn't keep hammers and chisels from being useful to sculptors.

The point is that the computer is a tool, just as a paint brush or a silk screen frame is a tool—no more and no less. To deny its utility to the artist is as senseless as treating it as a "thinking machine." The recent interest in this expressive medium is traceable to the fact that the cost of computer graphics systems has dropped to the point that they are showing up in hundreds of thousands of homes. The computer has become an affordable expressive tool that can be used effectively by any artist who is willing to invest a little time in learning how to use it.

You may believe that computers are "number-crunching" machines that require the constant attention of people who have exceptional technical skills. Although some computers fit this description, it doesn't mean that *all* computers are primarily mathematicians' tools or that *you* have to be a wizard to run one.

In fact, the inexpensive Texas Instruments 99/4A computer, with the Logo language, is a marvelously simple tool to use. Furthermore, it can be used to create graphic designs of exceptional beauty. All that is required on your part is that you be able to press keys on a keyboard and that you acquire the vocabulary of the Logo language. This book will provide the framework to allow you to learn this language easily and naturally.

Once you have mastered the computer and its language, you will have opened the door to a new creative medium—computer graphics.

There is one characteristic of computer graphics that you should know about before we start—it is different. It is different from watercolors, oils, pastels, block printing, weaving, pottery, and every other art form we have ever had. Because it is different, computer graphics will never replace any of these other art forms. Because it is different, computer graphics will let you express ideas in totally new ways—ways that are as far removed from your present art form as pen and ink is from carved marble.

So don't think I'm going to ask that you give up any expressive tool in your kit—I'm not. What I hope will happen is that you will find the computer to be useful in expressing ideas that aren't readily expressed by the methods currently at your disposal.

Computerists Anxious about Art

I've met many people who feel that they can't express themselves artistically. That's unfortunate, because humans have expressed themselves artistically since our ancestors found that images could be scratched on a cave wall with a rock. Of course, not everyone *wants* to be an artist, and that's fine. Not everyone wants to be a brain surgeon, either.

The problem is that many people who don't express themselves artistically think that they *can't* express themselves in this way. There is overwhelming evidence that this is not true.

Take a pen and a sheet of paper and draw a line on it. That wasn't hard, was it? Is the result "art"? Maybe not, but if you drew the "right" lines it would be. How does one draw the right lines? Regardless of stylistic differences, there is one thing that artists have in common. They not only look at the world around them, they *see* the world around them. The key to good artwork, then, is often good *seeing*.

The mind's eye—the imagination—is a marvelous tool for seeing. Can you imagine? Can you imagine an ice cream cone the height of the Empire State Building? Can you imagine the taste of the color green? What is the sound of red? How about the color of love?

Once you start imagining and seeing, you are ready to start expressing yourself creatively. If you are new to art, try many tools—crayons, paints, clay, pens, whatever you have available—even computers. After all, computers are useful artistic tools, and you have something to express.

Acknowledgments

Books don't appear, unassisted, on an author's desk; they are influenced by the support of many people. This book is no exception. The assistance of Texas Instruments was most appreciated, especially since this book was started just as new developments were taking place in the structure of the TI computer system and in the Logo language itself.

The freedom to create is the most powerful freedom of all. For their tolerance, acceptance, support, and encouragement in this task, I want to thank all those who helped me across the border from technology to art.

Finally, there is one for whom my thanks is greatest. May I always be helped to see the world through the eyes of love.

Los Altos, California D.T.
July 1983

Contents

I. Introduction

Computer Graphics—Here, There, and Everywhere

Computer-generated graphics are all around us. Businesses, long accustomed to typewritten documents, are now discovering that computer-generated plots of business data can be comprehended faster than data presented in a table. Educators, long-time fans of pictorial information, are discovering that computer graphics can enhance the teaching of subjects as diverse as logic and chemistry. And, of course, the entertainment applications of computer graphics are well known. To pick just one example, computer-driven video games illuminate millions of homes every day. The appeal of these games comes, in large part, from their colorful graphic images. Whether jumping frogs across a pond, jumping a man over a barrel, or smashing space bugs, the colorful animation holds the players' interest time and time again. Video game players are controlling their own cartoons!

Games are far from the only entertainment application of computer graphics, of course. The motion picture industry is now spending millions of dollars on massive computer systems to bring high-quality computer-generated images to the silver screen. In such films as *Star Wars*, *TRON*, and *Star Trek II*, the computer has established itself as a powerful ally for those who create video magic.

Beyond commercial applications, computer graphics systems have been used effectively as an expressive tool by many artists fortunate enough to have access to them.

Computers in the Home

Far from being the tools of a privileged few, computer graphics systems are available today for the cost of a clothes dryer. The personal computer industry grew from its modest beginnings a few short years ago to the point at which millions of computers are being purchased every year. The appearance of computers in homes means that millions of people now have access to technology that was once found only in industrial and academic environments. A graphic artist can now create computer-generated artwork in the privacy of a home or studio, instead of having to travel to a remote laboratory to gain access to this technology.

There are two major ways of generating computer graphics on a home computer system. One way involves the use of a *computer graphics program* that lets you use a graphics tablet or a joystick to "sketch" images on the screen. Using special commands in such a program, you might be able to fill outlines with color, rotate images, replicate parts of an image in other areas of the screen, and otherwise do many powerful things. The only problem with these programs is that they do not provide access to additional features that you might like to incorporate in your artwork. Because these programs are generated by other people, you must be content with these people's concepts of the graphic tools you need. Nonetheless, many of these programs are excellent and are appropriate for artists who feel the need to create artwork by moving a pen over a surface.

The second way to create computer graphics is to work directly with a computer language that is richly endowed with graphics commands. Certain computer languages have the property of allowing you to define your own procedures, which may be recalled by simply typing the procedure's name. This approach to computer graphics, *graphics programming,* brings you in closer touch with the raw power of the computer. It even allows you, should you wish, to create programs that do the things done by existing graphics programs, with the important difference that *you* are the one who determines which features to incorporate.

Contrary to popular belief, computer languages are not hard to learn. With the appropriate language, you can be creating interesting graphics programs in the first ten minutes of exposure to the system. One of the most appropriate languages for the creation of computer graphics is called Logo. The feature of Logo that makes it so powerful is called "turtle graphics."

The "turtle" is an imaginary creature residing on the display screen whose sole function is to obey your commands. By instructing the turtle to move in its present direction or to turn by some angle, you can create any image you desire on the screen. The turtle can be instructed to leave a trace of its path in any of several colors, to erase a line, and to obey a predefined series of commands, called a *procedure*.

Is the turtle easy to use? If you understand such words as FORWARD, BACK, RIGHT, and LEFT, you are well on the way to getting the turtle to create pictures on your television screen.

TI Logo

This book will focus on the powerful graphics environment provided by the Texas Instruments 99/4A computer, using the TI Logo or TI Logo II language cartridge. To make use of the material in this book, you will need the following equipment:

- TI-99/4A computer
- TI Logo or Logo II cartridge
- Memory expansion module
- Color (or black and white) television set or monitor
- Tape recorder cables and tape recorder, or disk drive interface and disk drive (for recording your artwork)

Depending on your needs and desires, you may also wish to record your artwork on film or videotape. In this case, you will need a camera with a tripod or a video cassette recorder. Fortunately, the cost of a video cassette recorder has fallen to the point that it is now a reasonably priced accessory for your computer system.

What We Will Do

The remainder of this book will deal with the following topics:

- Becoming familiar with the equipment
- Turtle graphics and the creation of static images

- The creation of animated shapes
- Designing animated sequences
- Projects in computer graphics
- Projects in animation
- Recording your artwork on film or on a video cassette recorder

We will take things step by step, and before long you will have mastered this tool. So set aside some time and prepare to enter the world of graphics programming. It is likely that you will find it fascinating.

II. Getting Started with Logo

Locating the Computer System

It is important that you find a good place to locate your computer system. Since you will be using it for creative applications, you should pick a place that is free of distractions. If you have a studio, find a quiet corner where the equipment won't be disturbed. At home, set up in a study or spare room away from heavy traffic flow. Your location should be as dust-free as possible and should be indirectly lighted. Find a tabletop that is large enough to comfortably hold the computer system and any related equipment you may be using (such as a video cassette recorder). Be sure you can reach the keyboard easily. Your creative ideas won't flow at all well if your arms have to strain to reach the keys! The television set or monitor should be placed where the screen may be seen easily without any strain on your neck or eyes. Be sure that no room lights reflect off the screen.

Although your computer space needn't be dark, it shouldn't be so bright that the screen colors appear washed out. Most important, the computer should be in a place you like—a place that is conducive to the free flow of ideas.

Making the Right Connections

I wish we could start making pictures right away, but in this regard computers are different from pencils and paper. Computers have to be set up and turned on. Fortunately, unless you move your equipment around a lot, your system will have to be set up only once. The manuals that came with your Texas Instruments 99/4A computer system show you how to set up all the equipment, so those instructions won't be repeated here. There are two types of equipment setups you could have, depending on when your computer was purchased. Computer systems purchased before mid-1982 had separate boxes for the memory expansion module and disk drive interface. Each box plugged into the adjacent one, so that the complete system resembled a train with the engine (the computer) connected to the memory expansion, which was connected to the disk interface, and so on. If your system is of this type, you should remember to turn the equipment on in the following sequence:

1. Turn on the disk drive and disk controller (unless you are going to use a tape recorder to store your graphic procedures).
2. Turn on the memory expansion.
3. Turn on the computer.

Of course, you should also turn on your TV set, or you won't be able to see your pictures.

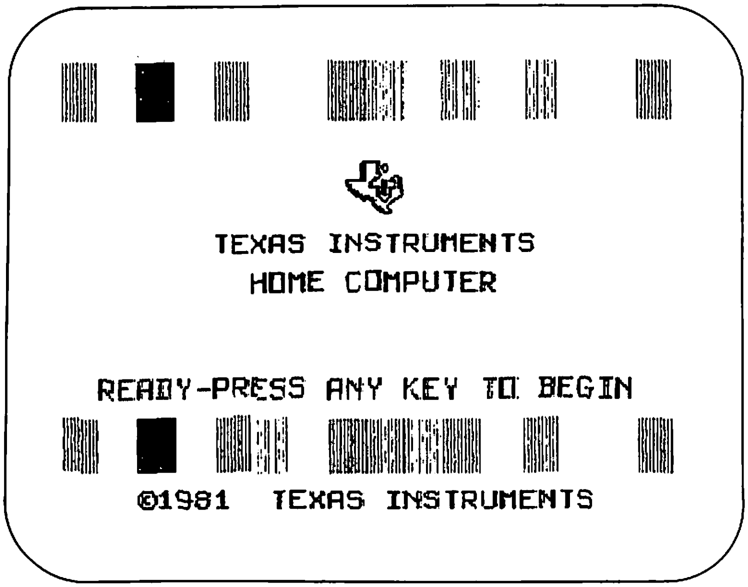
For computer systems generally available after mid-1982, the extra memory and disk drive are located in a single box placed behind the computer. To use this system, turn on the power to the peripheral box before turning on the computer itself.

You must remember to have your Logo cartridge plugged in before you start. Also, you should find the plastic strip that slides into the holder above the computer keyboard. This strip has the words DEL, INS, ERASE, CLEAR, BEGIN, PROC'D, AID, REDO, BACK, and QUIT printed on it. Some of the commands needed to control Logo use these words. To activate any of these keys, you will hold down the FCTN key while pressing the number key directly below the label of the word you desire.

Starting Logo for the First Time

If you have used Texas Instruments Logo before, feel free to skip this section. What we will do here is show what happens—and what can go wrong—when starting the Logo system.

First, turn on the power to your computer by itself. Do *not* turn on your memory expansion power first (this time). Once the TV has warmed up, you should see the following display.



The squares at the top and bottom are in a rainbow of colors. This is a great test pattern to use in adjusting the color balance of your television set. Once you have adjusted the colors of your set to your satisfaction, press any number or letter key on the keyboard. The screen will now show

```
TEXAS INSTRUMENTS
HOME COMPUTER

PRESS
1 FOR TI BASIC
2 FOR TI LOGO
```

When you press the number 2 on the keyboard (to start Logo), you will see the following message on the screen:

```
CHECK SYSTEM, PRESS 'BACK'
```

This message lets you know that the computer isn't working properly. The reason for this is that we didn't turn on the memory expansion unit. Logo is incapable of operating with only the memory contained in the main console. You must remember *always* to turn on the memory expansion unit before powering the computer itself.

Now turn everything off and start over, remembering to turn on the extra memory first. If you repeat our previous steps, you will now see the following message on the display:

```
WELCOME TO TI LOGO
?_
```

This is much better.

If you look at the upper left corner of the display, you will see a black flashing underscore sign (_). This is called the *cursor*. Its function is to let you know where a letter will appear when you type something on the keyboard. The question mark is a *prompt*. This symbol lets you know that the computer is waiting for you to give it an instruction.

You should take a few minutes to get familiar with the keyboard, since it is the principal tool through which you will convey your desires to the computer. Other ways include yelling at the system (not very effective), turning off the machine (effective, but final), and the use of joysticks (to be covered in a later chapter).

At first glance, the keyboard resembles that of a typewriter, at least so far as the placement of the letters and numbers is concerned. Some keys have two symbols on them. The letter R also has a left square bracket ([), the letter T has a right square bracket (]), the question mark is located on the I key, and so on. To type any of the gray characters on the front sides of the letter keys, press the key marked FCTN and hold it down while pressing the appropriate letter key. The FCTN key operates much like the SHIFT key on a typewriter. The SHIFT and CTRL keys on the computer keyboard also work in this manner.

Before using the keyboard, find the key marked ENTER. This key is pressed at the end of every line of instructions we give the computer. If we see an error before pressing this key, we can back up and fix it. Once ENTER is pressed, the computer will try to do the things you have entered in your commands. Because ENTER refers to a set of commands, this key should not be pressed at the end of each line of characters on the screen, unless the end of the line corresponds to the end of your instructions. If you are typing a long line of instructions, the computer will automatically continue printing the instructions on the next line if you run out of room. Only press ENTER when you are all done.

As an example, try typing the following:

```
THIS IS A TEST OF THE LOGO SYSTEM WITH LONG LINES
```

If you make a mistake while typing, just hold down the FCTN key and press the key under the word ERASE (the 3 key) until you have erased the wrong letter. Continue retyping to finish the line. You will see that as you typed SYSTEM the SY appeared on one line and the STEM appeared automatically on the next line. You do not need to press ENTER until you are finished with your line of commands. If you press ENTER now, you will find that Logo doesn't know what to do with your sentence—but perhaps that isn't too surprising.

You should note that Logo only uses uppercase characters. Since our main concern is graphics and not text, this should not present any problems.

A Brief Introduction to Logo

Now that we have come this far, let's get better acquainted with Logo. Type the following word and press the ENTER key:

```
HELLO
```

When you pressed ENTER the computer responded with

```
TELL ME HOW TO HELLO
```

As you can see, our greeting was not too well received. Whenever you see the message TELL ME HOW TO, it means that you have used a word that is not in Logo's vocabulary. Logo starts out with a vocabulary of words that lets it do many useful things. Even better than that, however, Logo lets you create definitions for words of your own choosing (we'll see how later). The fact that we can create our own definitions is why Logo's error message is encouraging us to "tell" Logo what we mean.

Perhaps this system is pretty friendly after all.

Introducing the Turtle

Of all the features of Logo, the turtle is central to computer graphics. As mentioned earlier, the turtle can be thought of as an object to which we send messages. These messages come either directly from the keyboard or as a result of letting Logo use a procedure we have defined.

The turtle draws its pictures on a portion of our display screen. Since the screen is entirely available for the display of text when Logo is started, we need to shift from the text display mode to the graphics display mode. To make this transition, we can send a message to the turtle. Type

```
TELL TURTLE CLEARSCREEN
```

and press the ENTER key. Remember, if you make a mistake while typing, hold down the FCTN key and press the key under the word ERASE (the 3 key) enough times to back up over the mistake; then retype the line.

As soon as you have entered this command, you will see a small black triangle in the middle of the screen and the flashing cursor at the left edge of the screen, roughly three-quarters of the way down. We have entered a mode in which we can see both the pictures drawn by the turtle and a few lines of commands as we enter them.

The turtle is shown as a triangle. The location of the triangle shows that the turtle is in the center of the screen, and its direction shows that it is pointing towards the top of the screen. As soon as we entered the words TELL TURTLE, Logo was instructed to pass all relevant messages to the turtle for execution.

What kinds of messages can we send to the turtle? Let's find out!

III. Making Drawings with the Turtle

We are now at a point where we can start to create pictures. In order to do this, we need to learn the words in Logo that make the turtle draw lines on the display screen.

Some Basic Commands

Before we can get any drawings on the screen, we must let Logo know that our commands are being directed to the turtle. The reason we must specify that the commands are to be followed by the turtle is that TI Logo lets us use some other turtle-like objects (called *sprites*) that also respond to turtle commands. (We will learn about sprites when we study animation in a later chapter.) By entering the words

```
TELL TURTLE
```

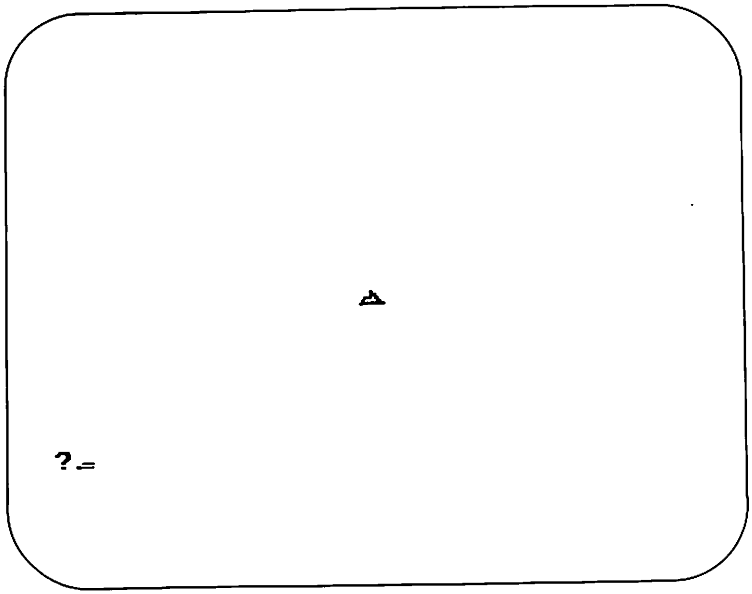
we instruct Logo to send all relevant messages to the turtle until we decide either to send them to another object or to return to the text screen. To turn the graphics off and return to a full text display, we use the command

```
NOTURTLE
```

We now want to explore the turtle's world a bit, so enter

```
TELL TURTLE CLEARSCREEN
```

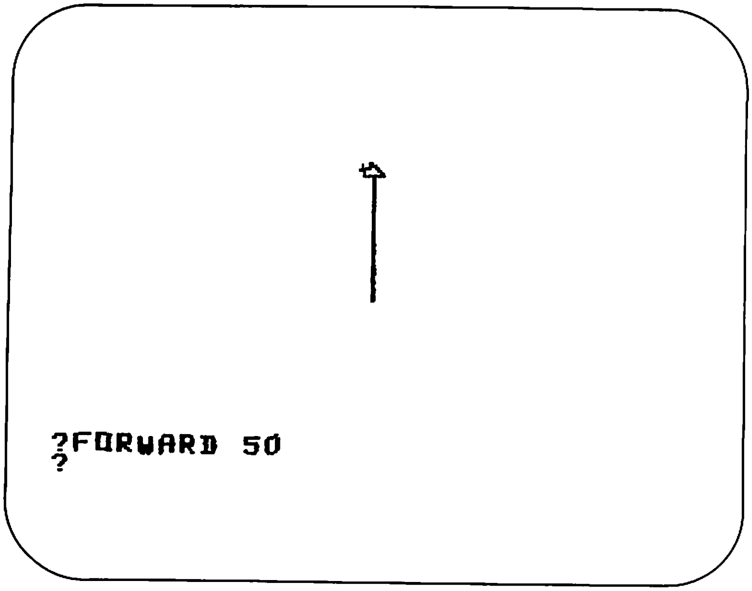
As soon as you pressed the ENTER key, your screen was divided into a graphics and a text window, and the turtle was placed in its "home" position and orientation.



Now let's find out how to move the turtle forward. We could try using the word FORWARD, but how much would the turtle move? This command must be followed by a specification of the number of screen units we want the turtle to move forward. Try entering

```
FORWARD 50
```

to see what happens.



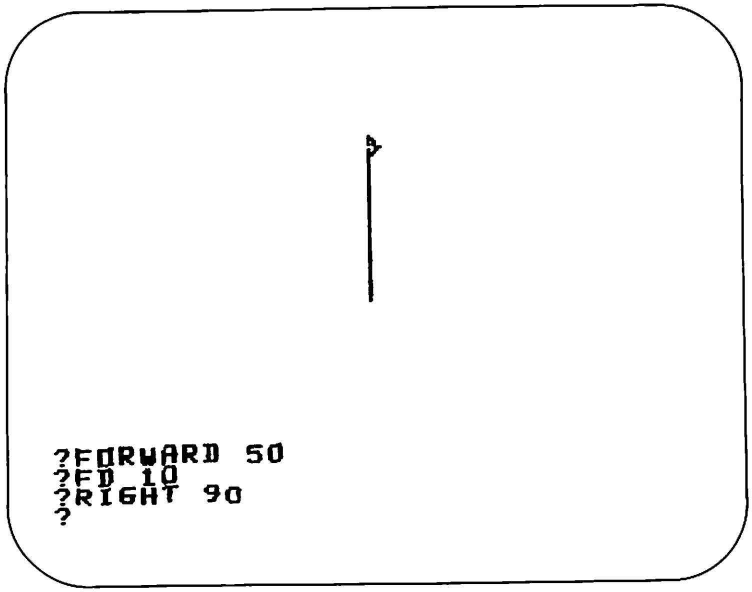
This command caused the turtle to move in its initial heading (toward the top of the screen) by 50 units, leaving a black trail behind. We can extend this line a little bit by giving the command

FD 10

The command FD is simply a shorthand form of FORWARD. There are short forms for many commonly used Logo words, and you will find that they save a lot of typing!

Unless we find some other commands soon, it is unlikely that the turtle will be very useful to us as a graphics tool. Fortunately, in addition to making the turtle move forward, we can also make the turtle change its orientation. To turn the turtle to the right by 90 degrees, for example, enter

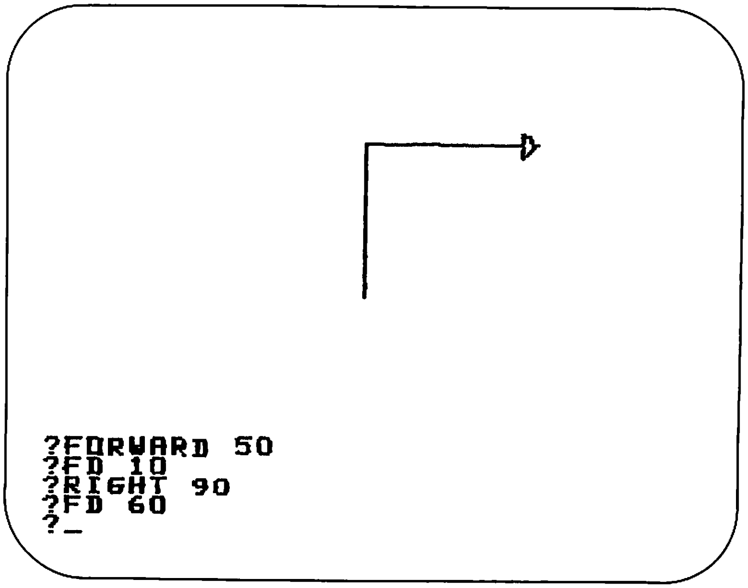
RIGHT 90



As you can see from the display, an advantage of having a visible turtle is that it is easy to tell in which direction it is pointing.

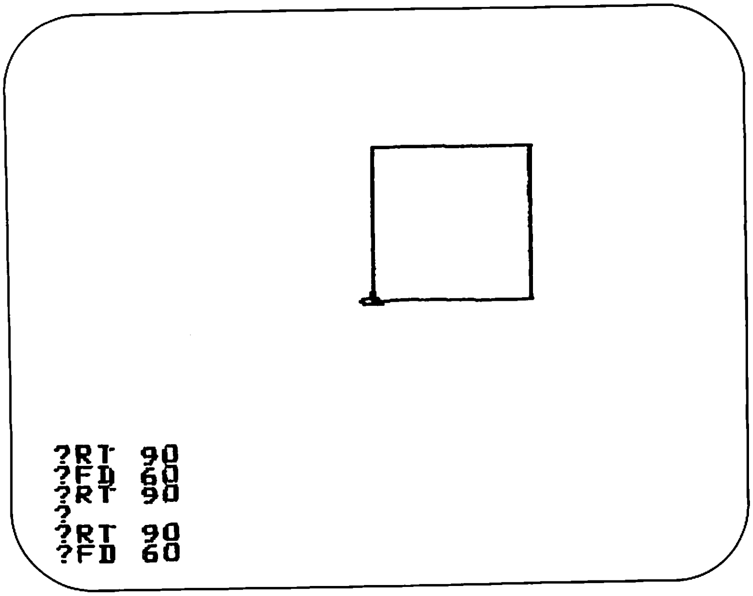
Let's see if we can use what we have drawn so far to draw a square. Since our first line is now 60 units long, we should move forward by this amount again. Enter

```
FD 60
```



The shorthand form for RIGHT is RT, so the following commands should let us finish the square:

RT 90
FD 60
RT 90
FD 60
RT 90



Notice, as you enter these commands, that the cursor automatically jumps to the top of the text window when it runs out of room at the bottom.

Why do you suppose we added the final RT 90 command, especially since the square appeared complete when we drew the last line? The final RT 90 command was needed to return the turtle to its exact starting position and orientation. As you gain more experience with turtle graphics, you will generally find it valuable to make sure each figure ends up with the turtle back at its starting position and orientation. This is especially true if you make pictures using simple figures as general building blocks that are used over and over again.

Does the square we drew look square on your display? It is possible that it appears a bit rectangular. Every TV set and monitor seems to have a different scale for horizontal and vertical lines. If you want to make squares appear square, you should adjust the vertical size control (usually located at the back of the TV) until the figure looks correct to you.

So far, we have learned how to use two very basic commands: FORWARD and RIGHT. If you feel adventurous, you might want to examine the use of the commands BACK and LEFT (or BK and LT). You should be able to determine quickly that BACK and LEFT work in opposite ways from FORWARD and RIGHT.

Repeating Commands

It may seem that you will have to do a lot of typing to make the turtle draw pictures. Fortunately, this is not the case. Logo has a command that makes it very easy to create figures that are generated from the repetition of commands over and over again (such as a square). This command consists of the word REPEAT, followed by the number of times the instructions are to be repeated, followed by a list of the instructions.

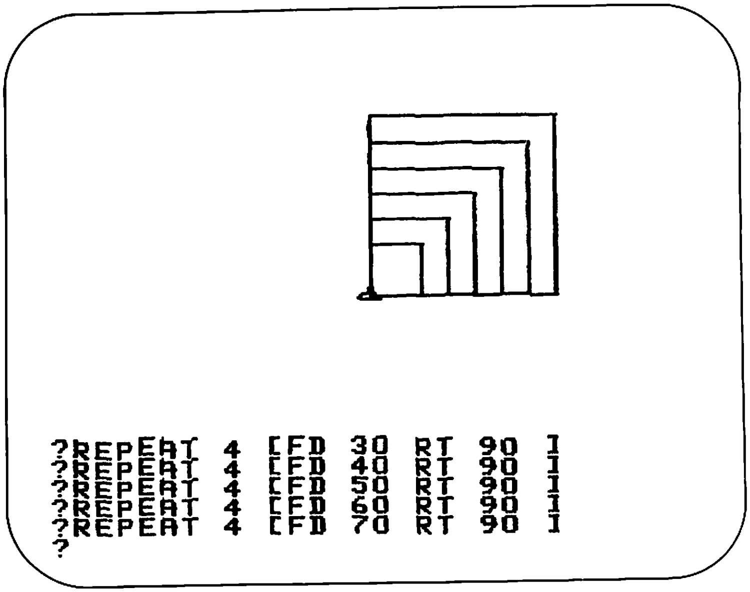
As an example, the eight steps we used to create a square can be replaced with this single command:

```
REPEAT 4 [FD 60 RT 90]
```

This command says, in essence, "Repeat, four times, the instructions 'move forward 60 units and turn right 90 degrees.'" To type this command, you need to use the square brackets ([]), which are found on the R and T keys. To type the left square bracket, hold down the FCTN key and press the R key. To type the right square bracket, repeat the process with the T key. Logo uses square brackets to enclose a *list*. Lists are collections of words that may be Logo commands or any of several other objects used by Logo. As we shall see later, the ability of Logo to work with lists is one of its strong points.

To try out our easy technique for drawing squares, enter the following:

```
CLEARSCREEN
REPEAT 4 [FD 20 RT 90]
REPEAT 4 [FD 30 RT 90]
REPEAT 4 [FD 40 RT 90]
REPEAT 4 [FD 50 RT 90]
REPEAT 4 [FD 60 RT 90]
REPEAT 4 [FD 70 RT 90]
```



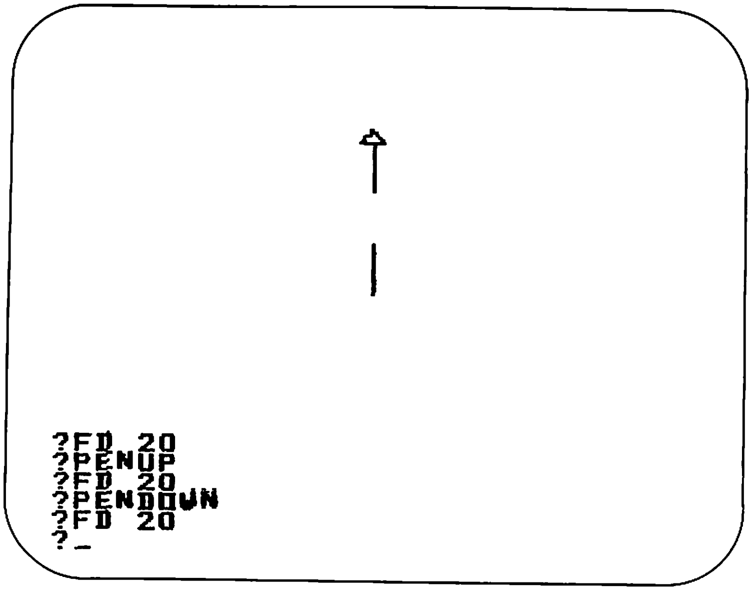
This set of nested squares was created with far fewer commands than we would have needed if we didn't have the REPEAT command at our disposal. We will see more creative uses of this powerful instruction as we progress.

Before generating any more pictures, we should learn more about things the turtle can do.

The Turtle's Pen

So far, we have seen that the turtle starts out with a black "pen" that draws lines as the turtle moves. There will be many times that we will want the turtle to move without drawing a line. For such times, it will be important to have a command that "picks the pen up." To see how Logo accommodates us in this regard, enter

```
CLEARSCREEN
FD 20
PENUP
FD 20
PENDOWN
FD 20
```



As you entered these commands, the turtle obeyed the instructions to lift the pen up and set it down again. The abbreviations for the PENUP and PENDOWN commands are PU and PD, respectively.

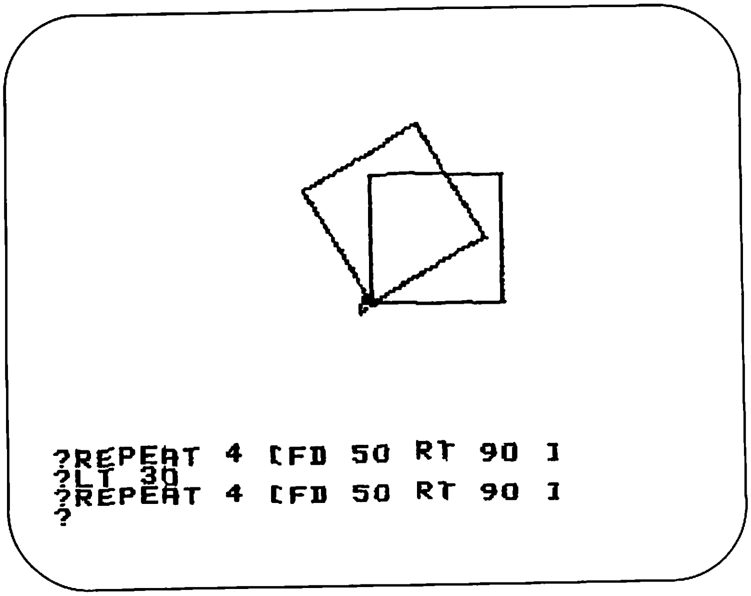
Now that you know how to move the turtle and have it lift and lower the pen, you might want to know how to erase a line you have drawn by mistake. To erase a line, we need to have the turtle carry an eraser instead of a pen. Once again, we will use an example to show how this is done.

Let's start by drawing a square (we will abbreviate CLEARSCREEN to CS):

```
CS
REPEAT 4 [FD 50 RT 90]
```

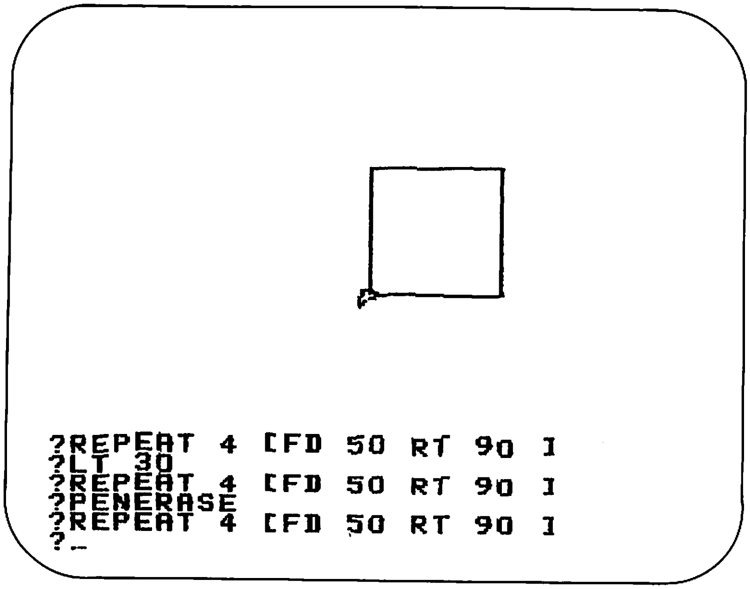
Next, let's add a second square, rotated to the left of the first one by 30 degrees:

```
LT 30
REPEAT 4 [FD 50 RT 90]
```



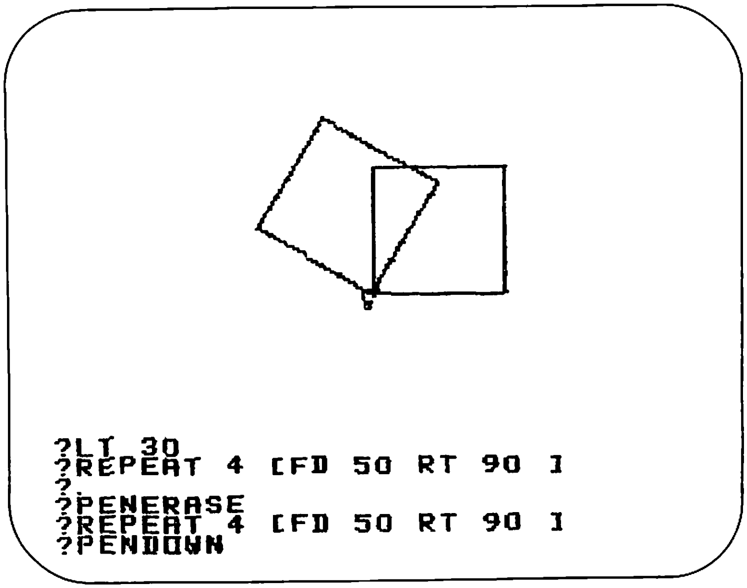
If we decide to erase this second square without disturbing the first one, we need only enter

```
PENERASE
REPEAT 4 [FD 50 RT 90]
```



To draw a new square rotated by 30 more degrees to the left, enter

```
PENDOWN
LT 30
REPEAT 4 [FD 50 RT 90]
```



The command PENERASE has the short form PE. To continue drawing after erasing some lines, just enter PENDOWN (or PD).

In addition to having an eraser, the turtle also has 16 pen colors from which you can choose. Each color has a number associated with it, as shown in the following table:

COLOR	NUMBER
Clear	0
Black	1
Green	2
Lime	3
Blue	4
Sky	5
Red	6
Cyan	7
Rust	8
Orange	9
Yellow	10
Lemon	11
Olive	12
Purple	13
Gray	14
White	15

To set the color of the pen you are using, enter the SETCOLOR command (abbreviated SC). The command

```
SETCOLOR 2
```

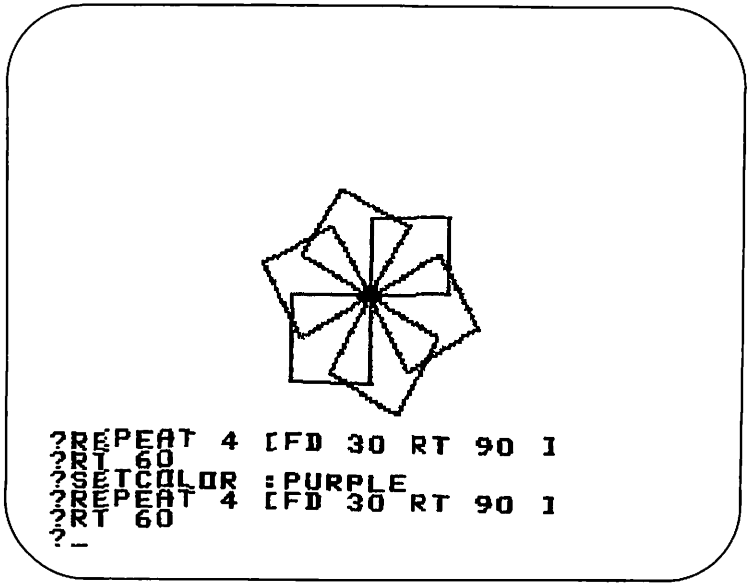
will set the pen color to green. If you have an aversion to using numbers, you will find that the command

```
SETCOLOR :GREEN
```

accomplishes the same thing. This works because each of the color words has been predefined by Logo to be the name of an imaginary box whose contents is the number corresponding to that color. In Logo, the contents of a named box is obtained by preceding the box's name with a colon (:). This convention will be described in more detail when we explore variables in the next chapter.

To see the effect of different pen colors, enter

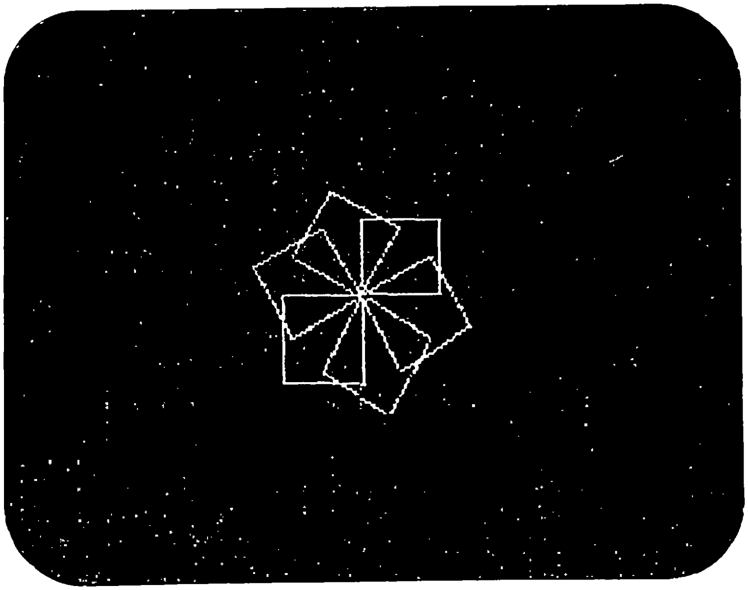
```
CLEARSCREEN
SETCOLOR :GREEN
REPEAT 4 [FD 30 RT 90]
RT 60
SETCOLOR :RED
REPEAT 4 [FD 30 RT 90]
RT 60
SETCOLOR :RUST
REPEAT 4 [FD 30 RT 90]
RT 60
SETCOLOR :YELLOW
REPEAT 4 [FD 30 RT 90]
RT 60
SETCOLOR :OLIVE
REPEAT 4 [FD 30 RT 90]
RT 60
SETCOLOR :PURPLE
REPEAT 4 [FD 30 RT 90]
RT 60
```



As you had the computer draw this colorful pattern, you probably noticed that where new lines crossed over lines previously drawn, the new lines picked up a bit of the previous lines' colors. This is an artifact of the way TI Logo draws lines, and it is unavoidable. Like many other unavoidable things, however, you can plan ahead and use it to your advantage in creating visually striking patterns.

Now that we know how to change the color of the turtle's pen, let's see how we can change the background color. To set the background color, we have to send a message to the background. We do this by typing, for example:

```
TELL BACKGROUND
SETCOLOR :BLACK
```



A black background really makes the turtle's pictures stand out! It also makes our text disappear, since we can't very well read black characters against a black background. To go back to our original background color, carefully enter

```
SC :CYAN
```

(Remember that you won't be seeing the letters as you type them. If nothing happens, you probably made a typing error. Press ENTER and try again.)

The Turtle's Limits

Now that we know as much about the turtle as we do, let's see how much room is available to us for drawing pictures. Enter

```
CLEARSCREEN
```

If the screen was cleared but the turtle didn't move to its home position and orientation, we must not have sent this message to the turtle. In fact, we sent it to the background instead. To remedy this, enter

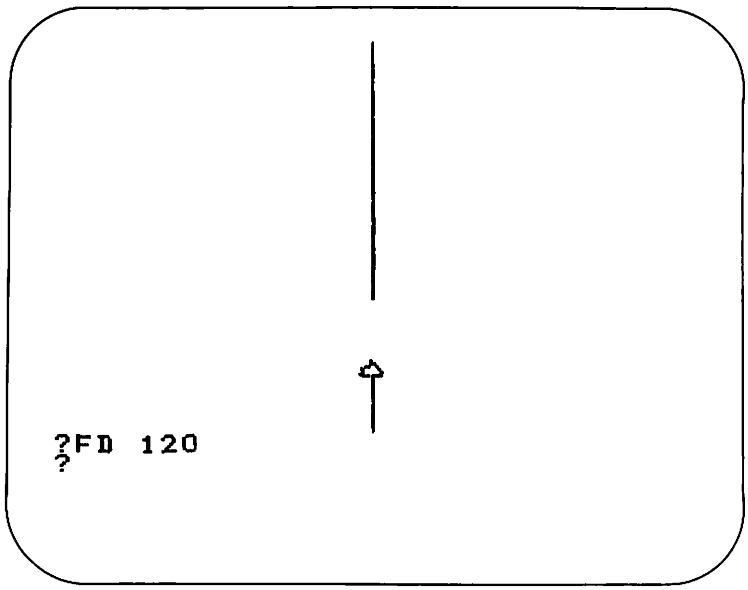
TELL TURTLE CLEARSCREEN SETCOLOR 1

That's much better!

Let's see what happens when we send the turtle on a long trip. If we enter

FD 120

we see a vertical line running off the top of the screen and reappearing at the bottom of the graphics window.



What happened?

When we tried to run the turtle off the screen, it "wrapped around" to the other side. This wraparound feature guarantees that you will never lose the turtle. It also means that you will have to be aware of the screen boundaries to make sure your pictures don't fragment at the edges.

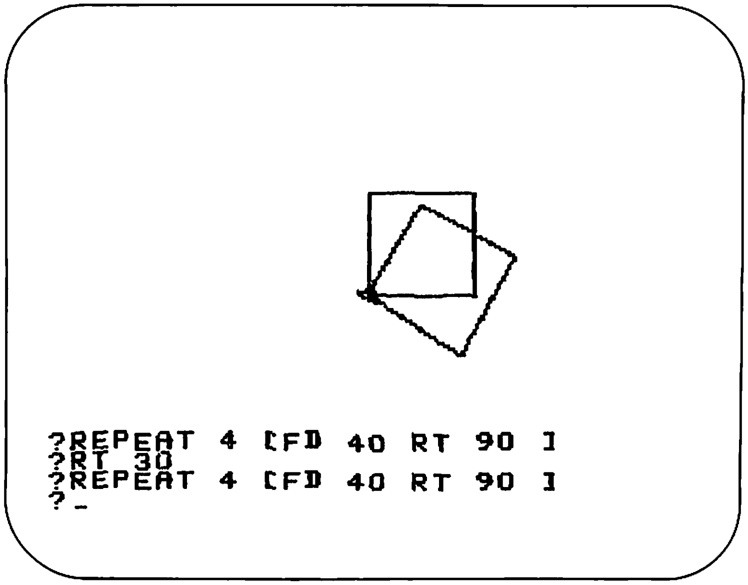
You should devise an experiment to see how far you can go from the home position without wrapping around. If you measure the extent of the turtle's world, you will find that it can move as much as 121 units to the left or right of the home position, as much as 98 units up from this position, and as much as 48 units below the home position.

If we call movements to the right or left of the home position movements along the x-coordinate, and movements above and below the home position movements along the y-coordinate, we can see that any point on the screen can be specified by a pair of numbers that give the x- and y-coordinates of the point.

Relocating the Turtle

Logo lets us pick the turtle up and place it at any location on the screen. This is extremely valuable if you want to start a picture somewhere other than at the center of the screen. Depending on the kind of motion we desire, we can use one of three relocation commands, SX, SY, and SXY. The letter S comes from the word *set*. These commands set the location of the turtle anywhere on the screen without drawing lines. To see how they work, enter

```
CS
REPEAT 4 [FD 40 RT 90]
RT 30
REPEAT 4 [FD 40 RT 90]
```



As you can see, these commands drew two squares, starting from the home position. The second square is tilted from the first by 30 degrees. Next, enter

```
SX -60
```

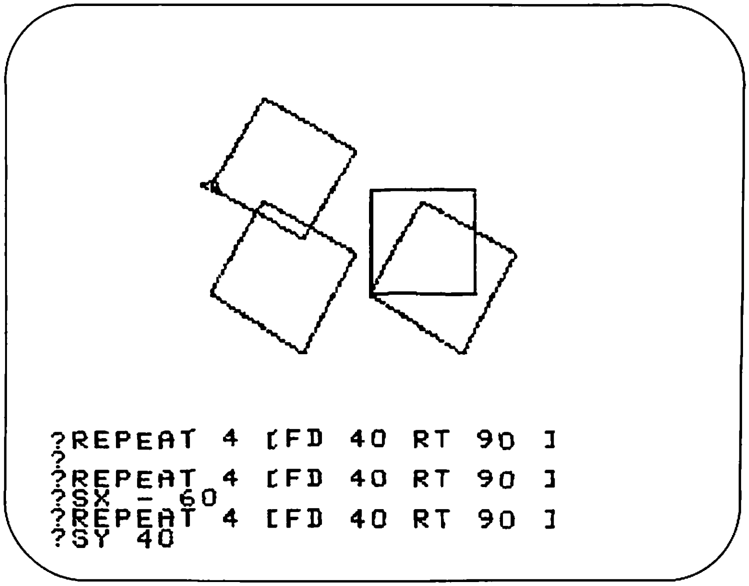
The minus sign (−) is obtained by holding the SHIFT key down and pressing the solidus (/) key. Notice that the turtle has moved to the left by 60 units. If you now enter

```
REPEAT 4 [FD 40 RT 90]
```

you will get a tilted square, starting at this new location. This shows that as the turtle is moved, its orientation is not changed in the slightest. Next, enter

```
SY 40
REPEAT 4 [FD 40 RT 90]
```

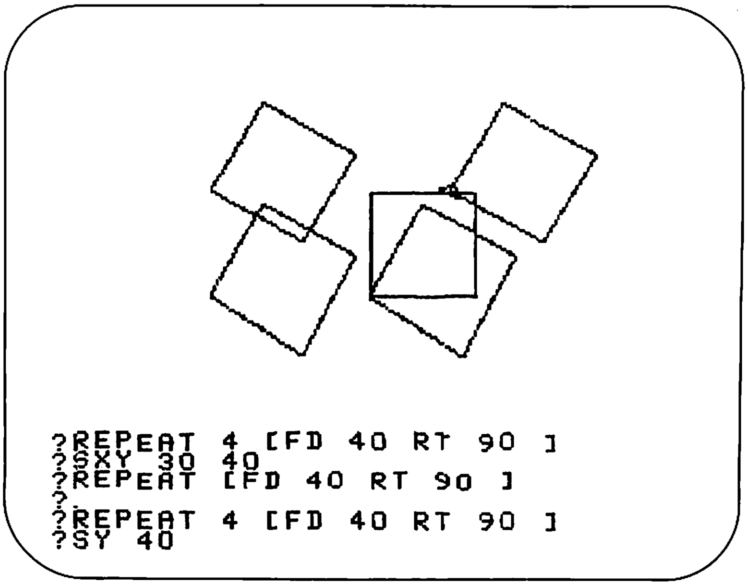
The turtle moved directly up from its previous location by 40 units and drew another square.



Now enter

```
SXY 30 30
REPEAT 4 [FD 40 RT 90]
```

The turtle moved to a point 30 units up and to the right of its home position before drawing another tilted square.



Finally, enter

SXY 30 −20

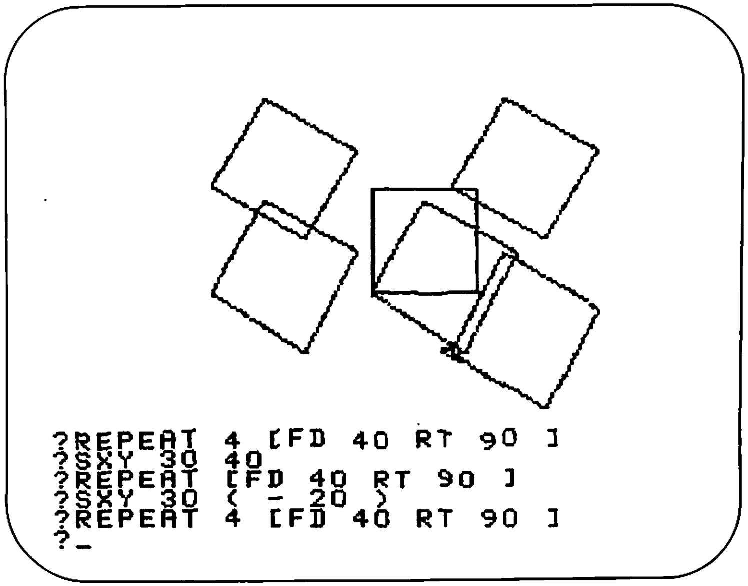
Oops—this gave us an error message:

TELL ME MORE

What we wanted to do was move the turtle to a location 30 units to the right and 20 units below the home position. We got an error message because Logo was trying to subtract the second number (20) from the first (30), and it didn't see the second number as the *y*-coordinate. The cure for this is to enclose the second number in parentheses if it is less than zero. As an example, enter

SXY 30 (−20)
REPEAT 4 [FD 40 RT 90]

to complete our picture.

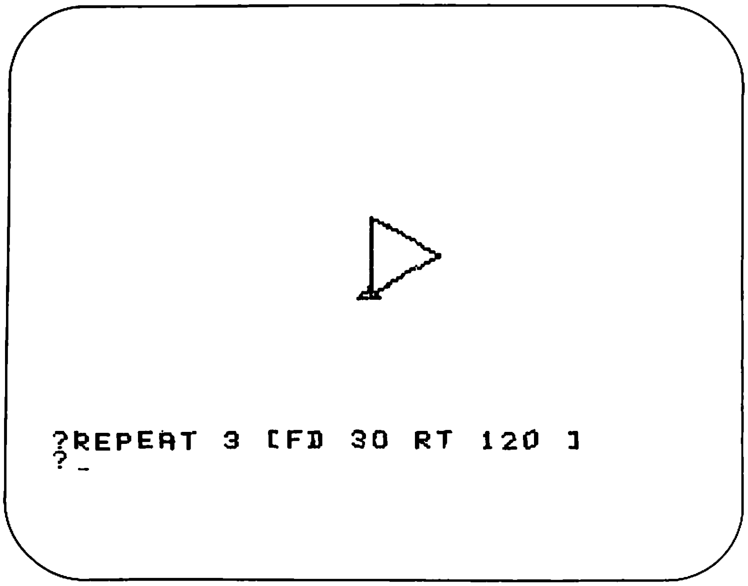


What we have found is that SX sets the *x*-coordinate of the turtle without changing the *y*-coordinate, SY sets the *y*-coordinate without changing the *x*-coordinate, and SXY changes both coordinates at once.

The ability to set the turtle's position is quite valuable. You can also set the turtle's heading to any value you want. This is done with the SETHEADING command (abbreviated SH). At first glance, SETHEADING looks similar to RIGHT, but these commands are very different from each other. To see why, enter

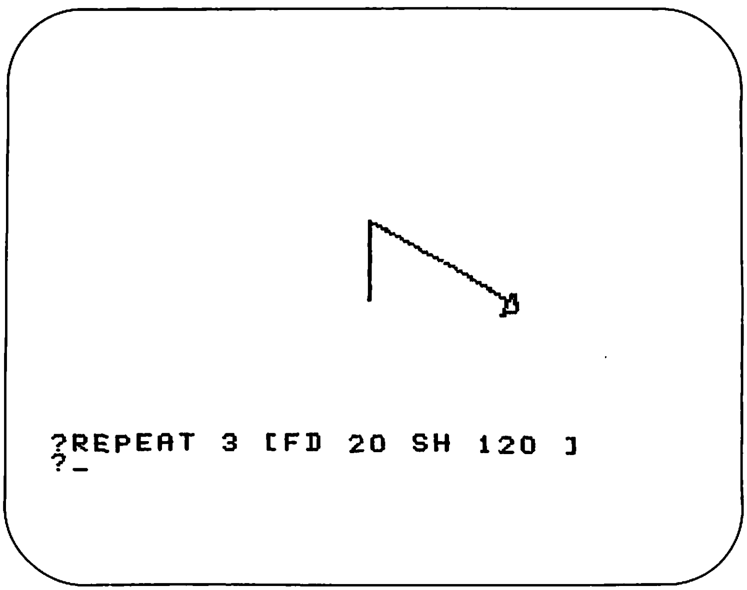
```
CS
REPEAT 3 [FD 30 RT 120]
```

This draws a triangle on the screen.



Next, enter

CS
REPEAT 3 [FD 30 SH 120]



This draws the first two sides of the triangle and then continues drawing the third side at an angle of 120 degrees. The commands RIGHT and LEFT turn the turtle by a specified amount from their previous heading. They are *relative* commands. The command SETHEADING turns the turtle *to* an angle measured from 0 degrees (pointing straight up). It is an *absolute* command.

Before finishing this section, you should know that you can make the turtle invisible by typing HIDETURTLE (or HT). There are two reasons you might want to do this. First, you may not want the image of the turtle to be part of your artwork. Second, the turtle draws lines much faster when it is invisible. To see the turtle again, just type SHOWTURTLE (or ST).

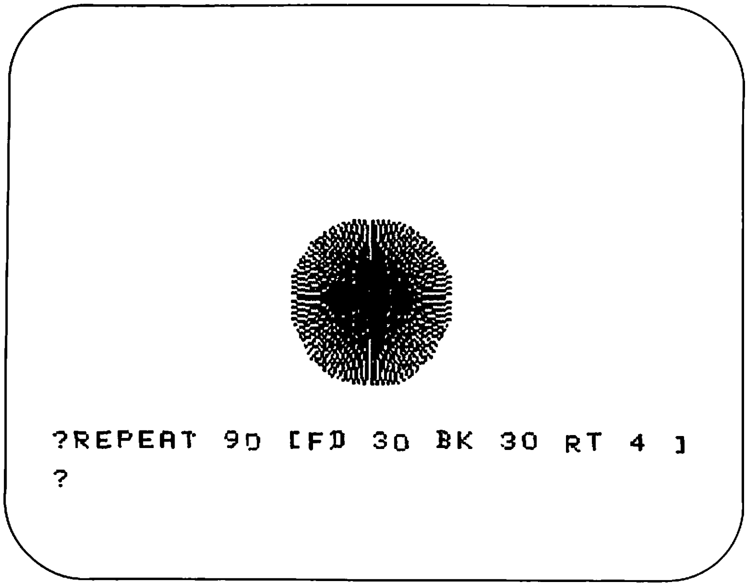
Some Art Projects

If you have worked straight through to this point, you deserve a rest. This chapter has given you many of the mechanical details of the Logo turtle, and it is now time to spend some effort using what you have learned. Before progressing to the next level, create some works of art to try out the skills you have obtained thus far.

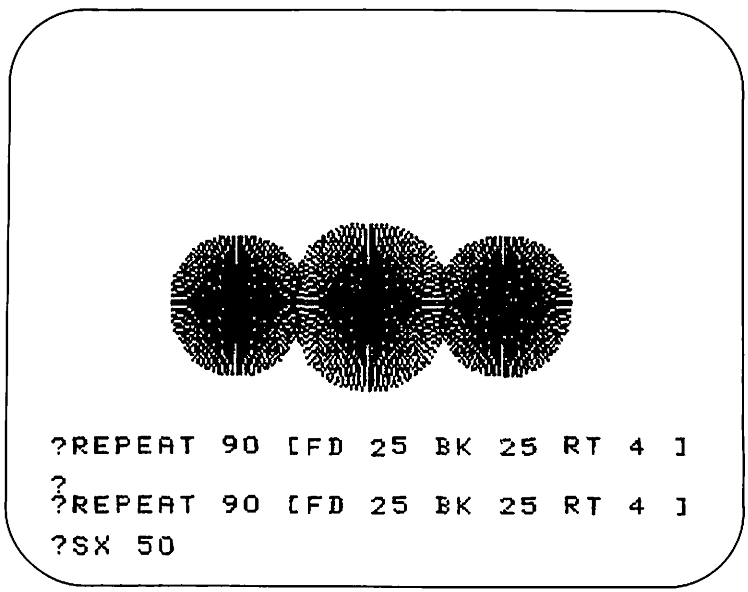
As an example, if you enter

```
CS SC :RED
REPEAT 90 [FD 30 BK 30 RT 4]
```

you will get this pleasant pattern.



How would you go about creating the following pattern?



One way to make this picture is to start with the previous one we drew and add the following commands:

```
SX -50
REPEAT 90 [FD 25 BK 25 RT 4]
SX 50
REPEAT 90 [FD 25 BK 25 RT 4]
```

Now you should experiment on your own. Try creating pictures that fit nicely on the screen. Then try creating pictures that use long lines (such as FD 300) to see the effects of wraparound. Set the turtle to some angle, such as 47 degrees, and send the turtle on a *very* long trip!

Play with pen colors. Play with background colors. Can you make the background flash from one color to the next?

As you experiment, you may see the error message

```
OUT OF INK
```

This message means that you tried to draw more lines than Logo could handle (we'll see why this happens later). This message doesn't mean that you did anything wrong, just that Logo ran out of graphics capabilities. Clear the screen and start another picture.

As you experiment, think about the texture of the graphics medium:

- Is computer graphics smooth or grainy?
- Is it hard or soft?
- Is it crisp or fuzzy?
- Which colors do you like?
- Does it seem that the computer lets you paint with light itself?
- Does the graphics screen seem like an electric loom, with the patterns woven in its surface?

You might want to keep a journal of your art projects and jot down your feelings about each picture you are creating. Later, it will be beneficial for you to reexamine your answers to these questions to see if your viewpoint has changed.

You should be pleased with your progress at this point. You are well on your way to mastering turtle graphics!

IV. Teaching the Turtle New Tricks

You have already learned enough about turtle graphics to let you create fairly elaborate pictures. Even some simple patterns still require a lot of typing, however, and all this typing can be quite bothersome, especially if you find that you want to use certain figures over and over again. Also, you have not yet learned how to save your handiwork for later viewing. We should learn how to create our pictures in a form that lets us save them for viewing on another day.

As in the preceding chapter, this chapter is primarily devoted to developing more skill with Logo. We will do a few art projects at the end; so if you are already familiar with the technique of defining procedures, saving and loading procedures on a diskette or cassette tape, and using variables, you may want to skip briefly through the first part of the chapter and then do some of the projects at the end. If you are not already familiar with these topics, this chapter will help you learn the remaining basic techniques needed to create stationary (as opposed to animated) works of computer art.

The Power of Procedures

Logo has a feature that lets us assign any collection of Logo commands to a word. As an example, instead of drawing a 30-unit square by entering

```
REPEAT 4 [FORWARD 30 RIGHT 90]
```

each time, we can create a *procedure* that draws a square each time we enter the word

```
SQUARE
```

When we create a procedure, the procedure's name (SQUARE, for example) joins the library of words that Logo knows about. The process of creating procedures lets you extend the Logo language to suit your own needs. Tailoring the language by adding your own procedures, each with a name of your choosing, lets all aspects of this computer graphics medium fit your own expressive style. Logo starts with just enough building blocks to get you started. From there on, you are on your own!

Defining Procedures

Let's see if Logo can draw a 30-unit square by using a single word. Enter

```
TELL TURTLE CS
SQUARE
```

The error message

```
TELL ME HOW TO SQUARE
```

lets us know that SQUARE is not currently part of Logo's vocabulary. To add this word to the vocabulary, enter the words

```
TO SQUARE
```

As soon as you press the ENTER key, the display screen will change color, and the display will show

```
TO SQUARE
END
```

Note that the cursor is located to the right of the word SQUARE.

By typing the words TO SQUARE, you entered the Logo procedure editor. This is a special part of Logo that lets us create and modify the definitions of words we want added to the Logo language. Every procedure definition has at least three lines. For the type of procedure we will create now, the first line consists of the word TO followed by the procedure name. The last line is the word END. The lines in between are the commands that make up the procedure. (We don't have any of these yet.) These commands can use standard Logo words, such as FORWARD, and the like, or they can use words for other Logo procedures you have created.

To define our SQUARE procedure, press the ENTER key and note that this opens up an empty line for us. Next, type

```
REPEAT 5 [FD 40 RT 144]
```

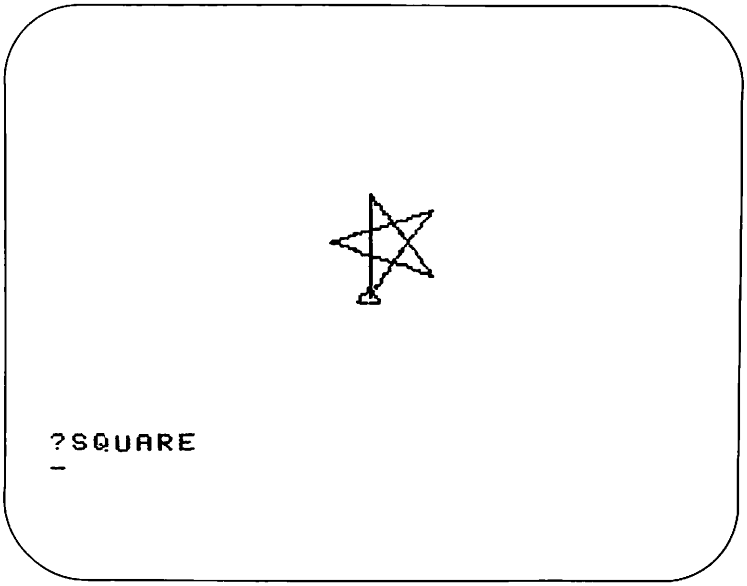
(Yes, I know this isn't going to give us a square; there is a method to my madness, so please bear with me.)

If you make a mistake while typing this line, you can fix it in the usual way by erasing the mistake with the ERASE key. (Remember to hold down the FCTN key when pressing the key under the word ERASE, and then retype the rest of the line.) When you have finished entering this line, you will want to return to that part of Logo that lets us draw pictures. To leave the editor, use the BACK key by holding down the FCTN key and pressing the key under the word BACK (the 9 key). This returns us to the part of Logo we were using just before we created the procedure; it also adds the definition of the word SQUARE to Logo's set of comands.

Let's see what we have accomplished. Enter

```
TELL TURTLE CS
SQUARE
```

This draws a beautiful—star!



Why did I have you intentionally create a procedure to draw a star instead of a square? There are two reasons. First, this exercise shows that Logo does not "understand" the meanings of your words. It just treats the words as arbitrary labels. This procedure would have drawn a star if it had been named CIRCLE, RECTANGLE, FRED, or A25X. So long as you don't try to use words that are already in Logo's vocabulary, the names you choose for your procedures are totally up to you. The second reason for misdefining SQUARE is that it allows us to learn how to use the editor to fix the procedure so that we will get a square when we type SQUARE.

To edit this procedure, type

```
EDIT SQUARE
```

When you press the ENTER key, you will see that we have reentered the Logo editor and that the SQUARE procedure is listed on the screen. To fix our procedure so that it will draw a 30-unit square, we must be able to move the cursor to the places we need to change and to insert and delete characters. The Logo editor has several keys to help us in this task. Each of these keys is activated by holding down the FCTN key and pressing the appropriate key, as shown in the following table:

KEY	FUNCTION
BEGIN	Move the cursor to the beginning of the line
PROC'D	Move the cursor to the end of the line
Up arrow (E)	Move the cursor up one line
Down arrow (X)	Move the cursor down one line
Left arrow (S)	Move the cursor to the left by one line
Right arrow (D)	Move the cursor to the right by one line
ERASE	Delete the character to the left of the cursor
DELETE	Delete the character above the cursor
CLEAR	Delete all characters from the cursor to the end of the line
BACK	Return Logo from the edit mode

To insert a character, you just type it, and the editor automatically shifts everything to make space for it.

Now, let's fix our procedure. First, move the cursor to the beginning of this line:

```
REPEAT 5 [FD 40 RT 144]
```

Next, move it (using the right arrow key) until it is under the 5:

```
REPEAT 5 [FD 40 RT 144]
```

Press DELETE (remembering to hold the FCTN key down):

```
REPEAT_[FD 40 RT 144]
```

and press 4:

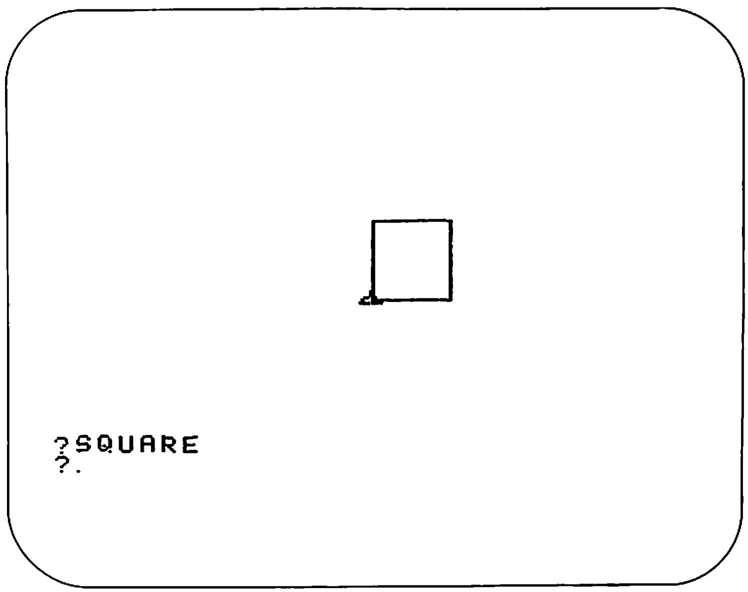
REPEAT 4_[FD 40 RT 144]

So far, so good. Now, continue to edit this line to change the 40 to a 30 and the 144 to a 90. When you are done, the line should read

REPEAT 4 [FD 30 RT 90]

At this point, press BACK, and Logo will have entered the modified definition of SQUARE into its library. Let's test our handiwork by entering

TELL TURTLE CS
SQUARE

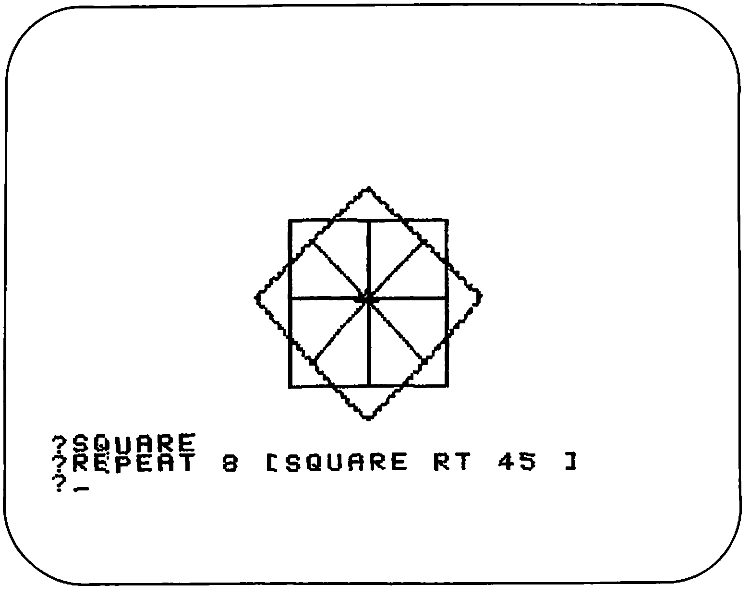


This time we see a square on the screen instead of a star.

Next, let's see if Logo treats SQUARE as it would one of its native words by entering the following:

```
REPEAT 8 [SQUARE RT 45]
```

When you press ENTER, you will see a pattern made from eight squares, repeated at equal angles around the center.



Now that you have learned how to define words in Logo, you can create complete works of art that can be displayed by typing only one word!

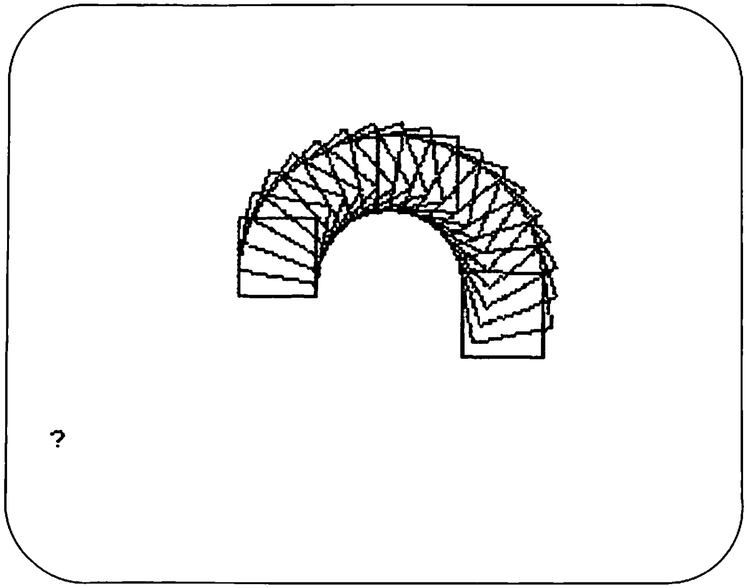
As an example, enter the following procedure:

```
TO PICTURE
TELL TURTLE CS HT
SX - 50
REPEAT 19 [SQUARE FD 10 RT 10]
END
```

To draw this picture on the display, just enter the word

PICTURE

and press ENTER.



Now that we have had a glimpse of how procedures can be useful to us, let's see how to save and recall procedures from the tape or disk memory. The advantage of saving your procedures is that you can recall them whenever you want without having to enter them by hand. You may also find that some of your procedures are powerful building blocks that you will want to use in many pictures. These building block procedures can be saved as a separate extension of Logo, which you might want to recall each time you turn the computer on to work on a new picture. We will soon see some examples of useful building blocks.

Saving and Recalling Your Logo Workspace

As you create new procedures, you are adding new words to Logo's vocabulary. Logo allows you to save these new words (and their meanings) in something called a *file* on cassette tape or on a flexible disk. In case you have forgotten the names of the procedures you have created, you can see their names by entering

```
NOTURTLE
PP
```

The command PP prints procedure names on the screen. Your screen should show the following procedure names:

```
TO SQUARE
TO PICTURE
```

To see the contents of any procedure, just enter PO (print out), followed by the procedure name—for example, PO PICTURE. If you want to erase a procedure from your Logo workspace, just type the word ERASE, followed by the procedure name (but don't do this yet).

When you save the procedures you have defined, you save all of them at once. Your file will contain the procedures for as many words as you have defined. Once your workspace has been saved on tape or disk, you may turn the computer off without fear of losing all your hard work. Every procedure you have defined (but not erased) will be saved on the tape or disk, ready to be recalled the next time you use the computer.

If you are using the TI disk system to save your procedures, you must first initialize a blank diskette, using the instructions that came with your disk system. If you are using cassette tape, you should save your procedures on high-quality cassettes (not the three-for-a-dollar variety). Follow the instructions that came with your computer system for connecting the cassette tape unit.

Once you are ready to save your procedures (that is, you have inserted the diskette in the disk drive and closed the door, or you have inserted the cassette tape in the tape recorder), type the word

```
SAVE
```

and press ENTER. Your screen will now show

```
      SAVE
PRESS     FOR
  1       PROCEDURES
  2       SHAPES AND TILES
  3       BOTH 1 AND 2

PRESS 'BACK' FOR TI LOGO
```

The only option that probably makes sense to you at this point is number 1, since we haven't yet learned about shapes and tiles (we will in a later chapter). Because you generally will want to save *everything* you have left in the workspace, however, you should press the 3 key. As soon as you do this, the screen will change to display

```
      DEVICE
PRESS     FOR
  1       CASSETTE
  2       DISKETTE

PRESS 'BACK' FOR TI LOGO
```

(There is a third option, which you will see if you are saving procedures only. This option will let you get a printed listing of your procedures if you have the TI thermal printer connected to your computer.) To save your workspace on cassette or diskette, press 1 or 2, as appropriate.

If you press 1, the screen will instruct you to rewind the cassette tape and press ENTER. Next, you will be instructed to press the RECORD and PLAY buttons on your recorder and press ENTER again. When the computer finishes saving your procedures on the tape, the display will ask if you want to check the tape to be sure everything was saved properly. Once you are finished, the computer will automatically return you to the familiar Logo display screen.

If you press 2 instead of 1, you will be saving your procedures on a diskette. The screen will change to show

```
TYPE FILE NAME, PRESS 'ENTER
  OR
PRESS
'SPACE' TO REVIEW FILE NAMES
  OR
'BACK FOR TI LOGO
SAVING -
NAME: ________
```

At this point you should pick a name of eight characters or less that describes your procedures. The reason for choosing a descriptive name is that your diskette ultimately will hold many of your procedure files. You could pick a name like SQUARES, for example, since both of our procedures use squares. You could also call it CHAPT4 to remind you that the procedures came from the fourth chapter of this book.

Let's type CHAPT4 and press ENTER. The light on the disk drive will turn on, and you will hear a whirring sound as the procedures are being saved. As soon as the computer has finished this task, the screen will return to whichever Logo display you were using when you entered the SAVE mode.

There is one important cautionary note if you are using the disk memory. You should always remember to remove the diskette when the power to the computer is being turned on or off. As a matter of general practice, you want to treat diskettes with great care and make sure that they don't get damaged by accidental "glitches" caused by power surges.

Now that you have saved your procedures, let's be bold and return Logo to its starting conditions. There are two ways to do this. You could turn off the computer and start over, or you could hold down the FCTN key and press QUIT. The latter technique has the same result as turning off the computer and turning it on again, but it is more gentle to the computer's electronic circuitry.

Once you have restarted Logo, type

```
PP
```

You will notice that there are no procedures in the Logo workspace. To recall the procedures you generated, type

```
RECALL
```

The display will now show:

```
     RECALL
PRESS     FOR
  1       PROCEDURES
  2       SHAPES AND TILES
  3       BOTH 1 AND 2

PRESS 'BACK' FOR TI LOGO
```

As in the SAVE mode, you should press 3. At this point, you will see a request to identify whether the data is on cassette or diskette.

If you select the cassette option, be sure the correct tape is loaded, and follow the directions shown on the screen. If you have problems loading your procedures from the tape, check to be sure the volume and tone settings on your recorder are set as described in the instructions that came with your computer system.

If you want to recall procedures from the disk drive, be sure the proper diskette is mounted in the drive, and press the number 2. At this point, you will be asked for the name of the file you want to have recalled. If you have forgotten the name of the file you want (or would just like to see the names of all the files on your diskette), press the space bar. The disk drive light will turn on, and you will hear a whirring sound as the computer locates the first file name. When the name CHAPT4 appears on the screen, you can recall the contents of this file by pressing the ENTER key.

After you press ENTER, the disk drive light will turn on again, and the procedures in the file called CHAPT4 will be loaded into your Logo workspace. After this has happened, the system will return to the familiar Logo display screen. To see that the procedures have been recalled, enter PP. You will see that the procedures SQUARE and PICTURE are now available for your use.

Because of the importance of saving and recalling procedures, you may want to review this section before proceding.

Variables

Although procedures such as SQUARE are quite useful, they have a limitation. Suppose you want to make squares of different sizes—perhaps one of 25, one of 37, and one of 43 units. We could create new procedures for each of these—SQUARE1, SQUARE2, and SQUARE3, for example. It would be far more powerful, however, if we could create one generalized procedure that would let us create squares of any size. The key to doing this in Logo is something called a *variable.* Before showing Logo's use of variables in procedures, we will make a slight digression to learn just what variables are.

A Logo variable can be thought of as a box that has a label and some contents. The box is named with a word, just as procedures are. When referring to the label for the box, the word is always preceded by a quotation mark ("). (The quotation mark is entered by holding down the FCTN key and pressing

P.) The contents of the box can be referred to by preceding the label name with a colon (:). Thus, the quotation mark denotes the name of the box, and the colon denotes the contents of the box.

What kinds of things can be stored in Logo variables? Variables can contain numbers, words, or lists. To explore this a bit, we will introduce two new commands: MAKE and PRINT. MAKE lets us create a variable and provide it with some contents. PRINT lets us print text on the display screen; we can use this command to examine the things inside Logo variables.

To try out these commands, enter

```
NOTURTLE CS
```

to clear the display screen. Next, enter

```
MAKE "HUEY 6
```

This command instructs Logo to create a variable whose name is HUEY and whose contents is the number 6. To see the contents of this variable, type

```
PRINT :HUEY
```

When you press ENTER you will see the number 6 on the display screen.

Next, let's enter

```
MAKE "HUEY "HORSEFEATHERS
```

If you now enter

```
PRINT :HUEY
```

the computer will display the word HORSEFEATHERS.

As a last example, enter

```
MAKE "HUEY [COW EAGLE DOG CAT]
```

and type

```
PRINT :HUEY
```

The computer will display the list COW EAGLE DOG CAT.

As you can see, we can make Logo variables hold all sorts of things.

Suppose we want to increase the value of a number that is stored in a Logo variable. How can we do this? Let's look at an example. First, create a variable that contains a number:

```
MAKE "FRED 23
```

If we type

```
PRINT :FRED
```

we will see the number 23 on the screen. Next, enter

```
MAKE "FRED :FRED + 1
```

What do you think this means? The best way to translate this command may be: "Make the variable whose name is FRED contain the contents of FRED plus one." We can see if we get the desired result by entering

```
PRINT :FRED
```

You will see that the number printed on the screen is 24, just as we expected. By repeating the last MAKE command (by using the REPEAT command, for example), we can increase the number in FRED to 25, 26, 27, 28, and so on. We have made a *counter*! Counters are very useful tools, as we will see later.

Now we are ready to examine the use of variables in procedures. As mentioned earlier, it would be very useful for us to be able to define a procedure that lets us create squares of any size. By typing SQUARE 27, for example, we could draw a 27-unit square.

To define this procedure, enter

```
ERASE SQUARE
```

to eliminate our old procedure, and then use the procedure editor to enter

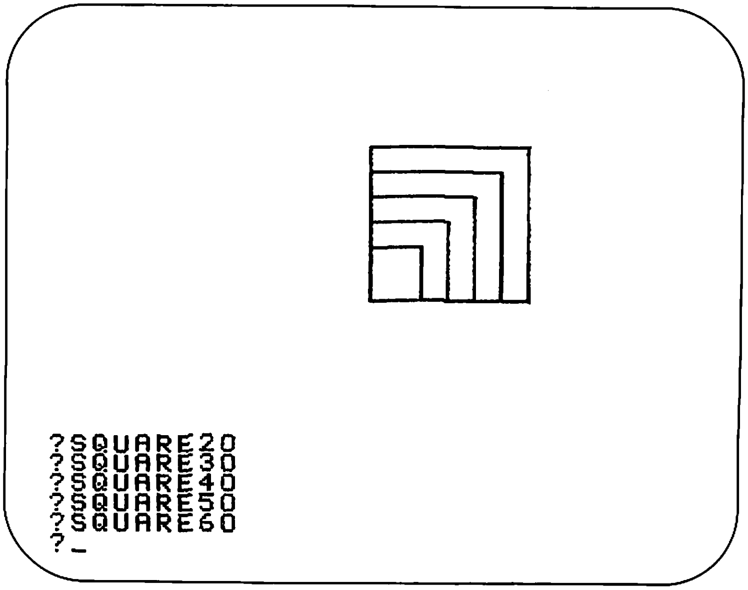
```
TO SQUARE :SIZE
REPEAT 4 [FD :SIZE RT 90]
END
```

To create this procedure, type TO SQUARE. When you press ENTER, you will see that you have entered the Logo procedure editor, with the cursor just to the right of the word SQUARE. Press the space bar once and enter :SIZE. Now press the ENTER key. Type the next line of the procedure, and press BACK to return to Logo.

This procedure uses the variable SIZE to contain the length of each side of the square. This variable is uniquely identified by its name and by the procedure in which it appears. This means that you could use the word SIZE as a variable in another procedure, and Logo would never be confused as to which variable you were using. If we give a command such as SQUARE 20, the local value for SIZE (20) will be "passed" inside the procedure to be used with the FD command.

Let's see how our new procedure works. Enter

```
TELL TURTLE CS HT
SQUARE 20
SQUARE 30
SQUARE 40
SQUARE 50
SQUARE 60
```



This is a much easier way to create nested boxes than the way we did it in the last chapter!

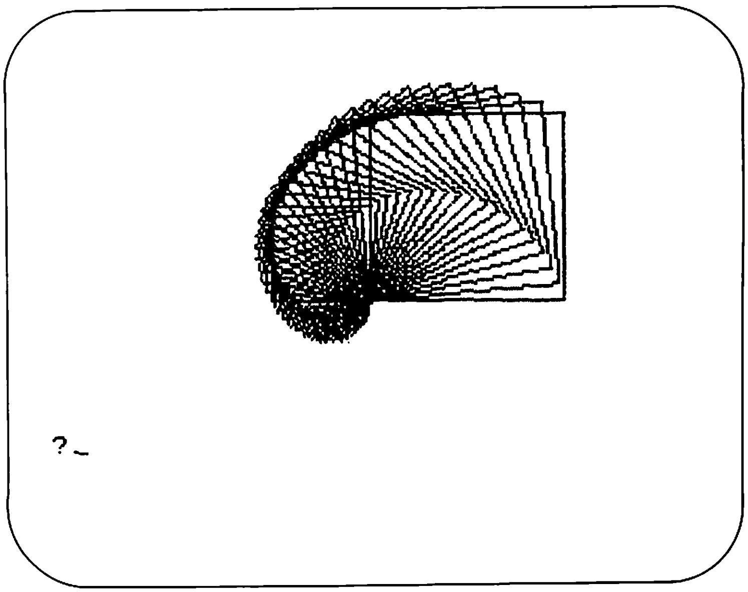
We can pass the contents of variables as well as numbers with our procedures. To see an example of this, enter the procedure

```
TO PICTURE2
TELL TURTLE CS HT SH 180
MAKE "COUNT 1
REPEAT 37 [SQUARE :COUNT RT 5 MAKE "COUNT :COUNT + 2]
END
```

After pressing BACK to leave the editor, enter

PICTURE2

This procedure creates a series of tilting squares, with increasing sizes. Each square's size is given by the contents of the variable COUNT, and this value is increased by 2 each time the command sequence is repeated.



Each time you create a new picture, you should save your workspace on tape or diskette so that nothing will be lost if you accidentally turn off the computer.

Projects that Use Procedures

Recent reports from the distant planet Spork indicate that their artists enjoy using a computer graphics language called GOLLI. The golli is actually a Sporkian animal that can frizzle in a straight line or can rizzle by up to 360 drigs before heading in the direction it started from. We found that one could draw a 40-unit square (called a *snarp* in GOLLI) by giving the command

```
SPLEGO 4 [FRIZZLE 40 RIZZLE 90]
```

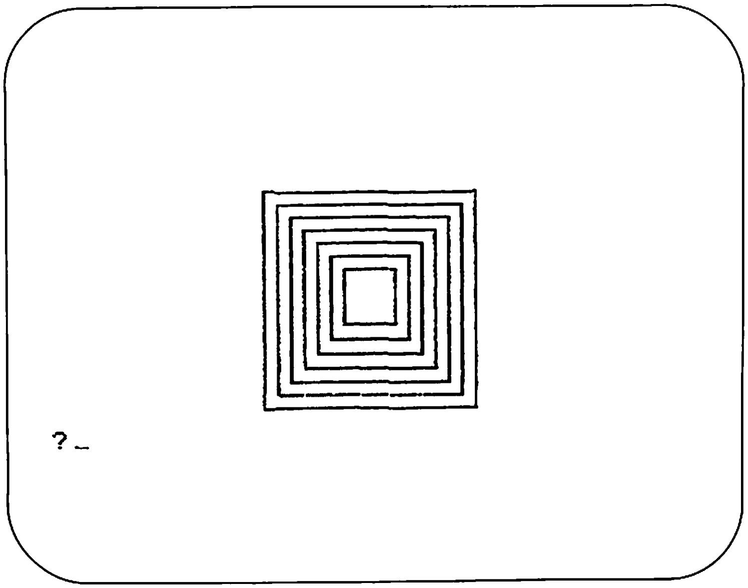
Your task is to create the procedures in Logo that let you enter this Sporkian command and have it work. Next, you should write the GOLLI commands to draw a five-pointed star and demonstrate that it works. Here is one procedure to help you get started:

```
TO SPLEGO :NUMBER :LIST
REPEAT :NUMBER :LIST
END
```

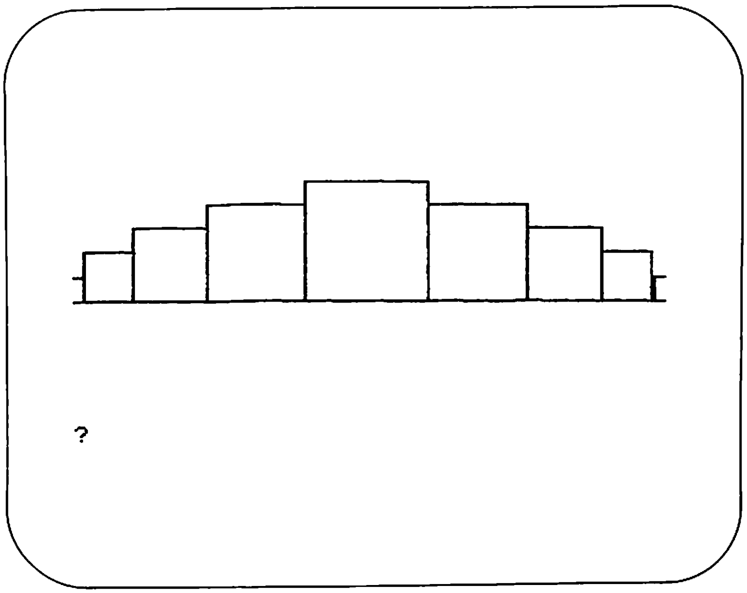
You should be able to create the other procedures you need without much trouble.

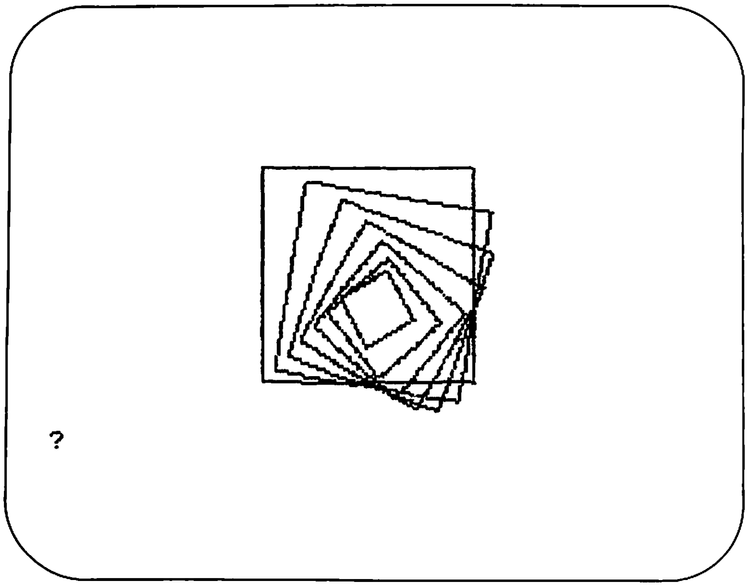
Moving back to more familiar terrain, you can create many interesting pictures with figures as simple as the square. As an example, the following figure was created with the procedure

```
TO PICTURE3
TELL TURTLE CS HT
SXY -40 (-40) SQUARE 80
SXY -35 (-35) SQUARE 70
SXY -30 (-30) SQUARE 60
SXY -25 (-25) SQUARE 50
SXY -20 (-20) SQUARE 40
SXY -15 (-15) SQUARE 30
SXY -10 (-10) SQUARE 20
END
```



See if you can create the procedures to generate pictures that look like those in the following figures.





The following procedures show one way to create these pictures:

```
TO PICTURE4
TELL TURTLE CS HT
SX -125
MAKE "SIZE 10
REPEAT 5 [SQUARE :SIZE RT 90 FD :SIZE LT 90 MAKE "SIZE
  :SIZE + 10]
MAKE "SIZE :SIZE - 20
REPEAT 4 [SQUARE :SIZE RT 90 FD :SIZE LT 90 MAKE "SIZE
  :SIZE - 10]
END
```

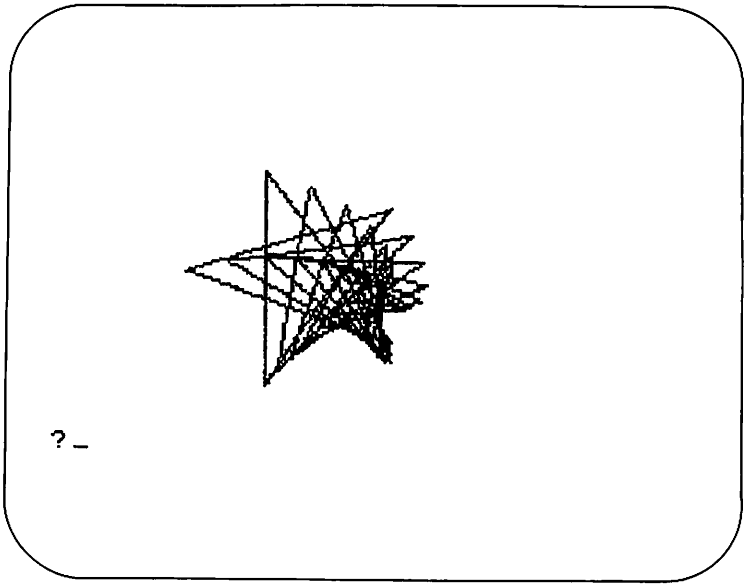
```
TO PICTURE5
TELL TURTLE CS HT
SXY -40 (-30) SQUARE 80 RT 10
SXY -35 (-25) SQUARE 70 RT 10
SXY -30 (-20) SQUARE 60 RT 10
SXY -25 (-15) SQUARE 50 RT 10
SXY -20 (-10) SQUARE 40 RT 10
SXY -15 (-5) SQUARE 30 RT 10
SXY -10 0 SQUARE 20 RT 10
END
```

Now you should create some pictures of your own that use squares. Try using different colors for the lines and background. Make pictures with a lot of symmetry; then make pictures that use random sizes, orientations, and starting positions. Which patterns do you like better? Write down your feelings about each picture in your journal. Save all your artwork for later viewing.

Finally, modify the procedure SQUARE so that it draws a five-pointed star. You will want to edit it to look like this:

```
TO SQUARE :SIZE
REPEAT 5 [FD :SIZE RT 144]
END
```

Now repeat the various PICTURE procedures we have created. Here is what PICTURE5 looks like with stars instead of squares.



Does the star change the nature of the overall picture by a little or a lot? Which patterns do you like better?

In the next chapter we will use our knowledge of Logo turtle graphics to create many more interesting patterns. You have now mastered the fundamental technique of turtle graphics, so let's move on to the next stage!

V

Four Little Polygons and How They Grew

Thus far, we have been able to create pictures using two kinds of building blocks—lines and squares. In the next few chapters, we will explore many classes of building block procedures that may be powerful tools in your hands. Some of the building blocks are quite primitive, and others are very detailed. Some of the detailed ones can stand on their own as designs, yet any of them may be used in combination with others to create even more beautiful patterns.

As in previous chapters, we will finish with a few projects to let you try out your skills. Before working on this chapter, you should clean out your workspace so that it contains no procedures. The easiest way to do this is to restart Logo by pressing QUIT.

Polygons

In the last chapter we used this procedure:

```
TO SQUARE :SIZE
REPEAT 4 [FD :SIZE RT 90]
END
```

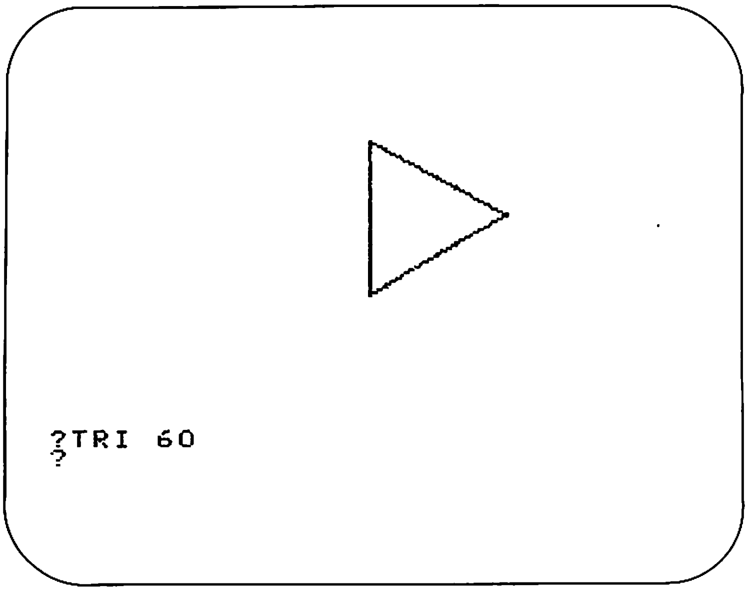
The square generated by this procedure is an example of a regular polygon. Although squares can form the basis for many interesting pictures, there also are many other pretty polygons to use. To see one example, let's create a procedure that generates an equilateral triangle:

```
TO TRI :SIZE
REPEAT 3 [FD :SIZE RT 120]
END
```

If you now enter

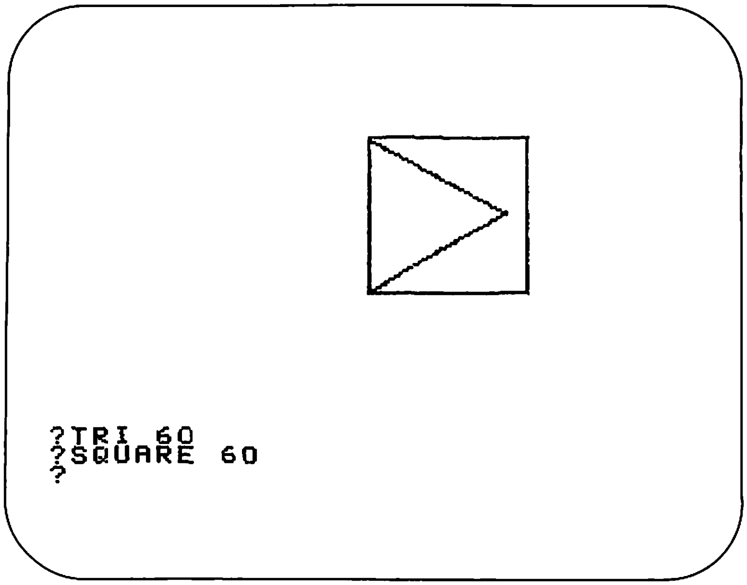
```
TELL TURTLE CS HT
TRI 60
```

you will see a triangle on the screen.



You can draw a box around the triangle by entering

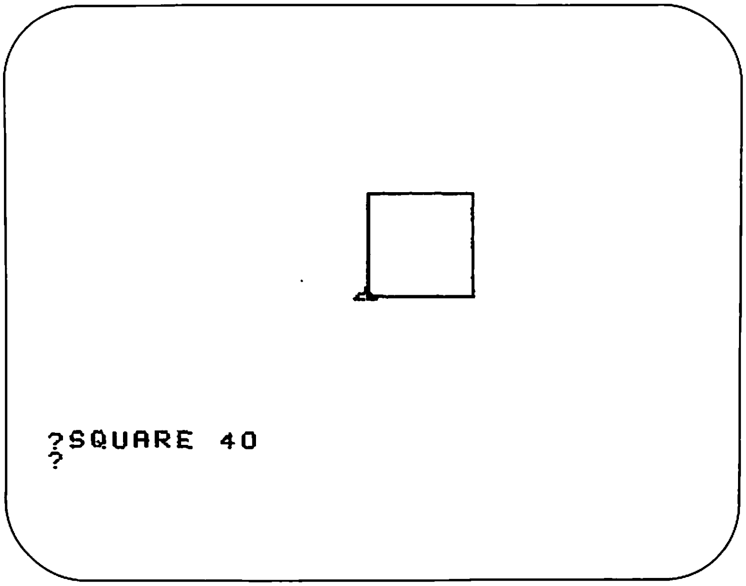
```
SQUARE 60
```



Suppose you wanted to draw a simple picture of a house. You could use a square for the front of the house and a triangle for the roof. Experiment a bit to see how to put the roof on the house.

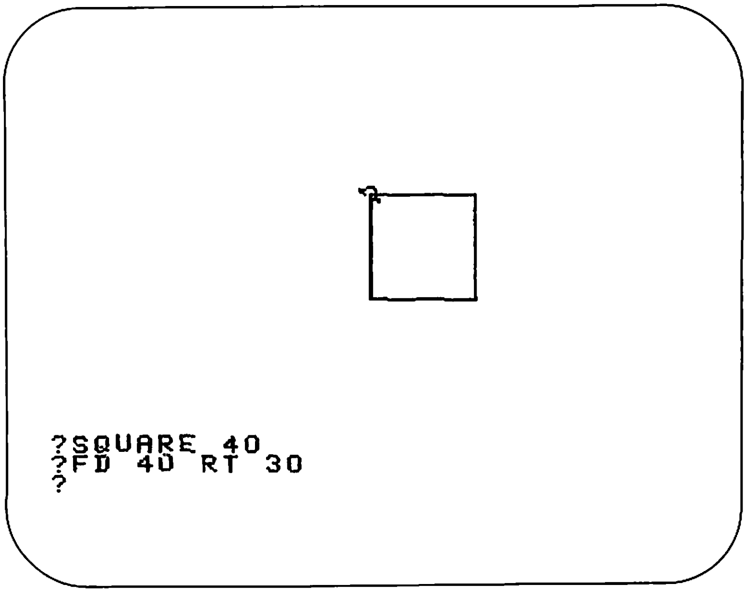
One way to create the house is to start with the bottom part:

```
CS ST
SQUARE 40
```



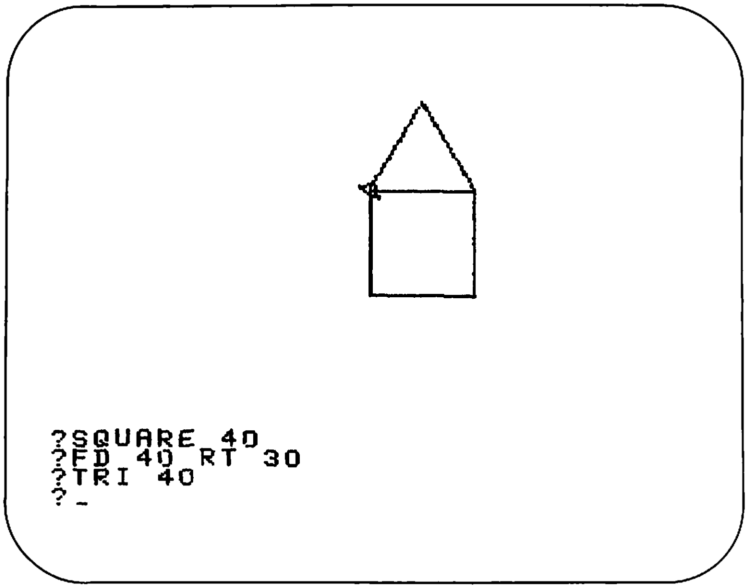
and then move the turtle to the upper left corner of the box and tilt it to the right by 30 degrees:

FD 40 RT 30



Finally, we can draw the roof by entering

```
TRI 40
```



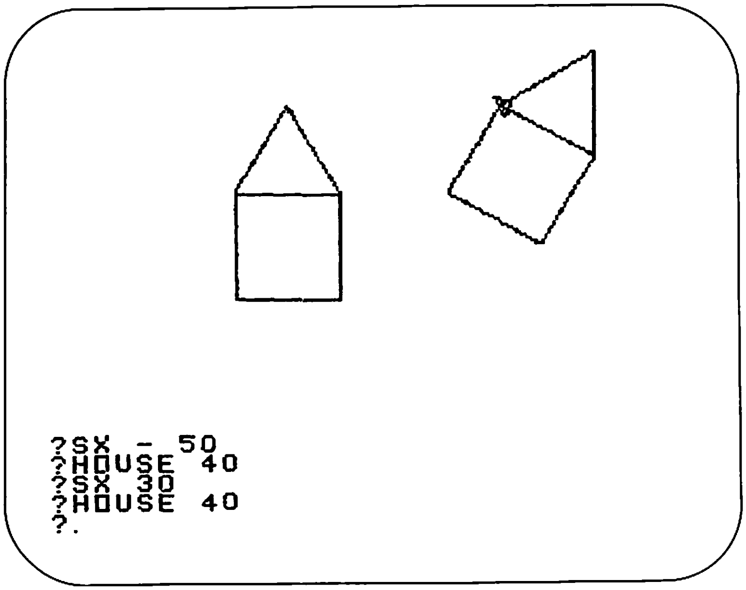
(As you can see in this exercise, the visible turtle can be of great help in checking on the placement of figures in our pictures.)

You could easily create a procedure for drawing houses of any size; for example:

```
TO HOUSE :SIZE
SQUARE :SIZE
FD :SIZE RT 30
TRI :SIZE
END
```

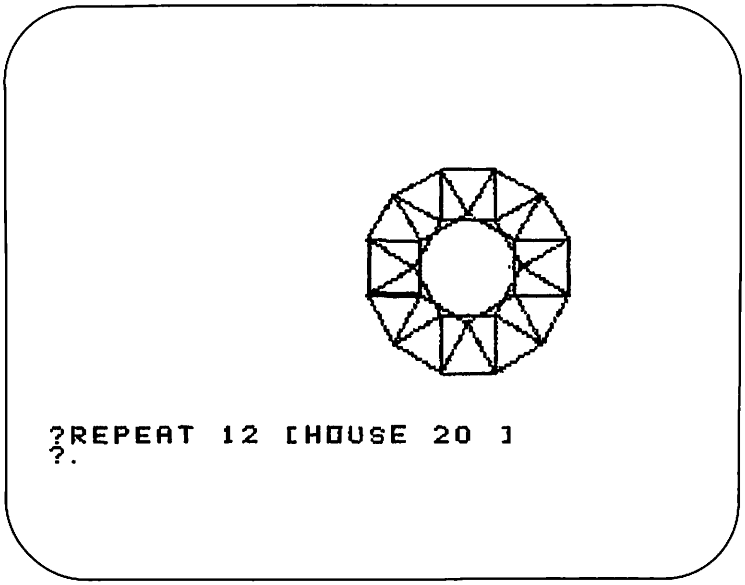
Let's use this procedure to create two houses side by side:

```
CS
SX -50
HOUSE 40
SX 30
HOUSE 40
```



Oops—our second house is tilted and offset from the first. This happened because the procedure HOUSE does not leave the turtle in the same position and orientation it had when it started. Before we fix this problem, let's experiment with HOUSE some more. Enter

```
CS HT
REPEAT 12 [HOUSE 20]
```



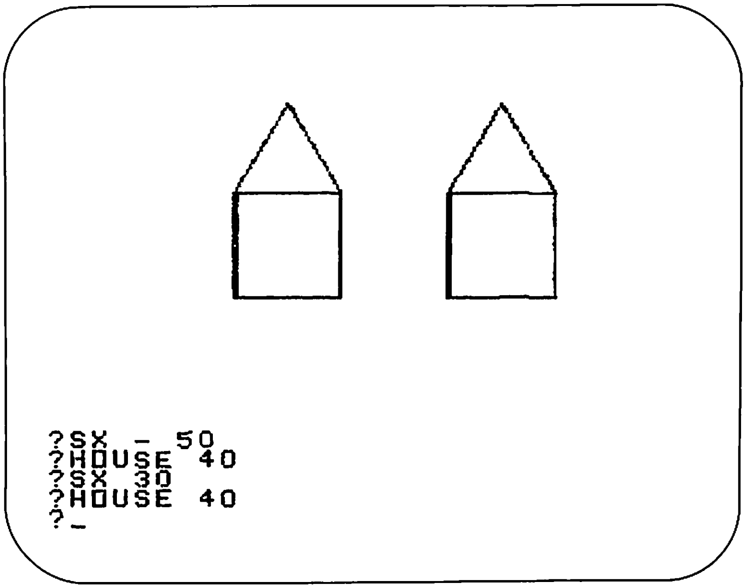
This is a pretty pattern—one you might wish to develop into a more detailed picture, using different sizes and colors for HOUSE.

To fix our problem with HOUSE, we can return the turtle to its starting position and orientation by adding one line to the procedure. Modify the HOUSE procedure so that it looks like this:

```
TO HOUSE :SIZE
SQUARE :SIZE
FD :SIZE RT 30
TRI :SIZE
LT 30 BK :SIZE
END
```

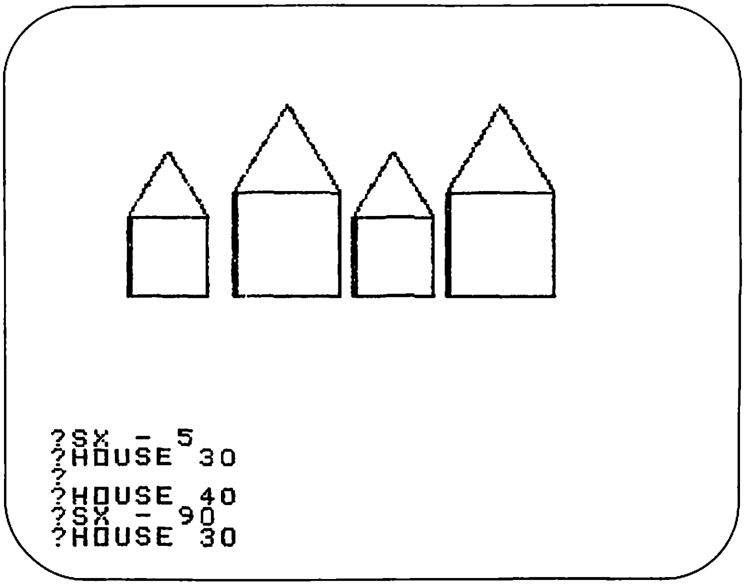
Now, if we repeat our previous experiment, we will get two houses side by side:

```
CS
SX -50
HOUSE 40
SX 30
HOUSE 40
```



We can add more houses of different sizes to this picture. Here are two more to get you started:

```
SX -90 HOUSE 30
SX -5 HOUSE 30
```



Blossoms

Triangles and squares are useful for much more than drawing houses. We can also create patterns by repeating polygons after turning the turtle by a chosen angle. If we keep repeating this process, we will get a flower blossom. As an example, in the last chapter we created a pattern by entering

```
REPEAT 8 [SQUARE 30 RT 45]
```

Suppose we wish to make a blossom-drawing procedure that lets us use any polygon we want as the repeated unit in the blossom. To do this, we need to create a list, such as [SQUARE 20 RT 45] or [TRI 30 RT 45]. We can put the polygon procedure name in a variable such as LIST, but we need to find a way to append RT 45 to this name. Logo lets you combine two lists with the SENTENCE (or SE) command.

See if you can figure out what SENTENCE does in the following procedure:

```
TO BLOSSOM :LIST
REPEAT 8 SENTENCE :LIST [RT 45]
END
```

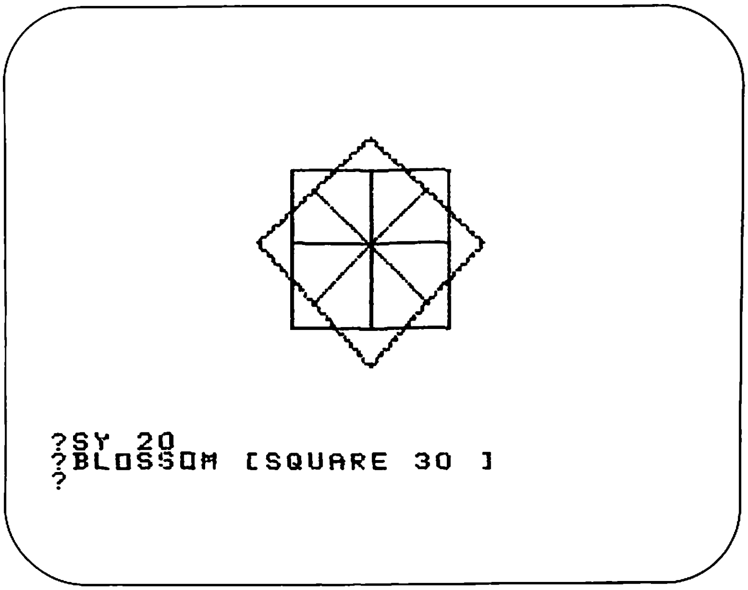
If the contents of LIST is [TRI 37], for example, SENTENCE creates a new list that looks like this:

```
[TRI 37 RT 45].
```

This is exactly what we want!

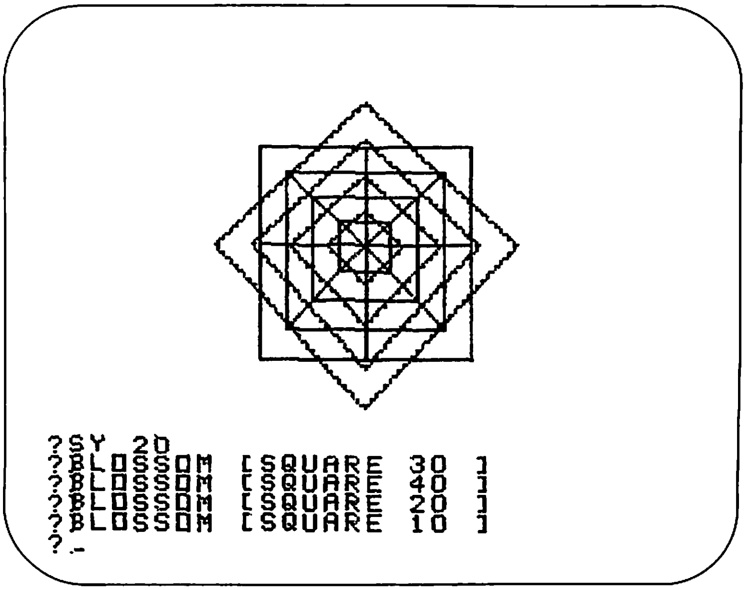
Try entering

```
CS HT
SY 20
BLOSSOM [SQUARE 30]
```



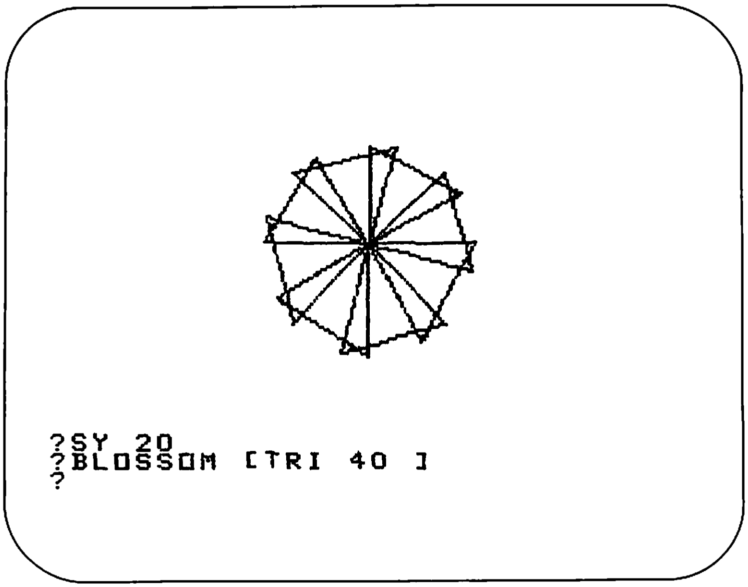
This gives us the same figure we saw in the last chapter. Now let's add to this picture by entering

```
BLOSSOM [SQUARE 40]
BLOSSOM [SQUARE 20]
BLOSSOM [SQUARE 10]
```



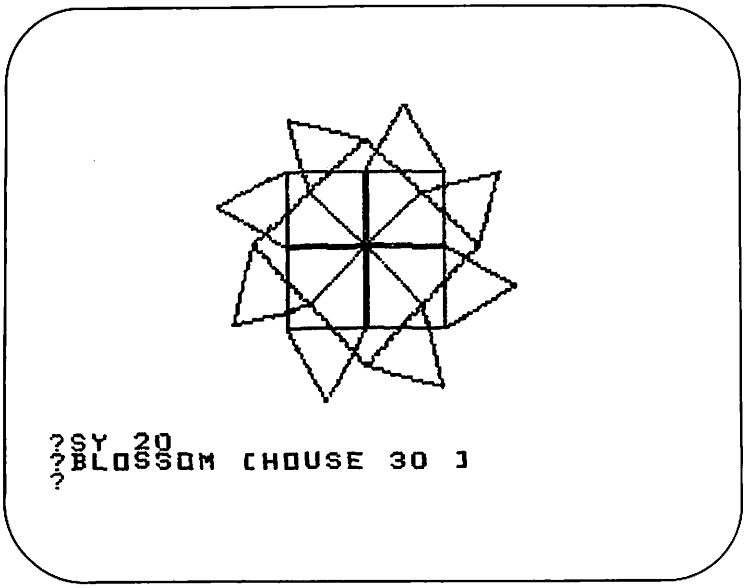
BLOSSOM can create many interesting patterns. Try

```
CS
SY 20
BLOSSOM [TRI 40]
```



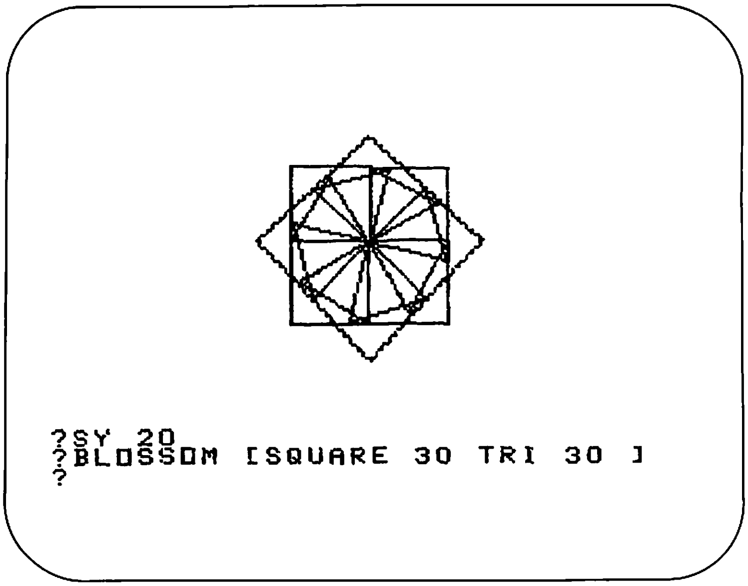
And try

```
CS
SY 20
BLOSSOM [HOUSE 30]
```



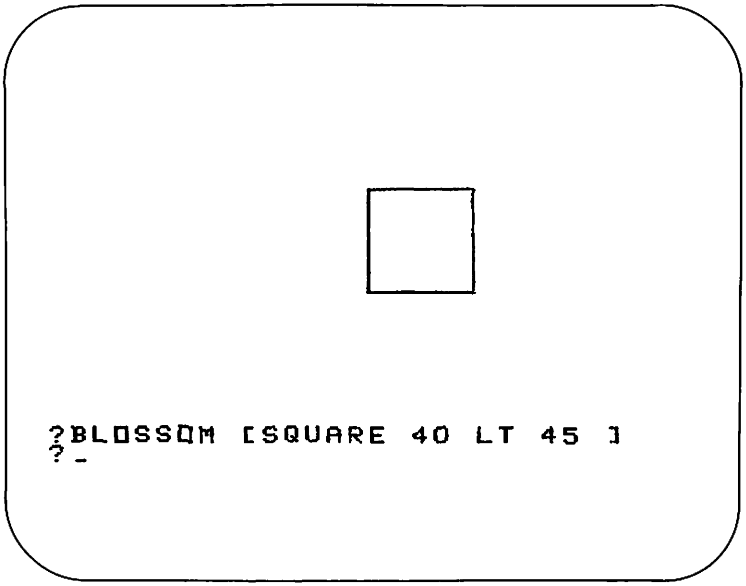
What do you think will happen if you try

```
CS
SY 20
BLOSSOM [SQUARE 30 TRI 30]
```



Were you surprised? If not, try to guess what will happen if you enter

```
CS
BLOSSOM [SQUARE 40 LT 45]
```



All we see is a single square! The reason for this is fairly simple—our command LT 45 acted to counterbalance the RT 45 command in the BLOSSOM procedure.

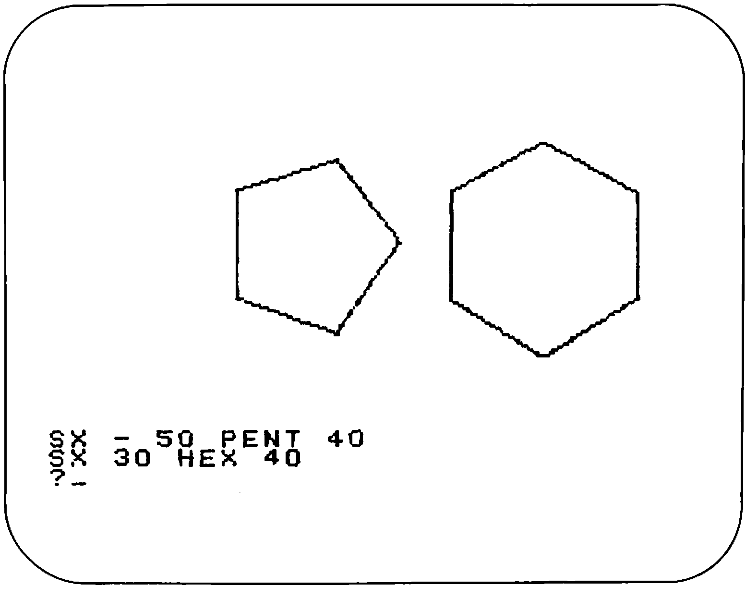
Now that we know about squares and triangles, what about other polygons? The following two procedures let you create regular pentagons and hexagons:

```
TO PENT :SIZE
REPEAT 5 [FD :SIZE RT 72]
END

TO HEX :SIZE
REPEAT 6 [FD :SIZE RT 60]
END
```

To see the figures generated by these two procedures, enter

```
CS HT
SX -50 PENT 40
SX 30 HEX 40
```

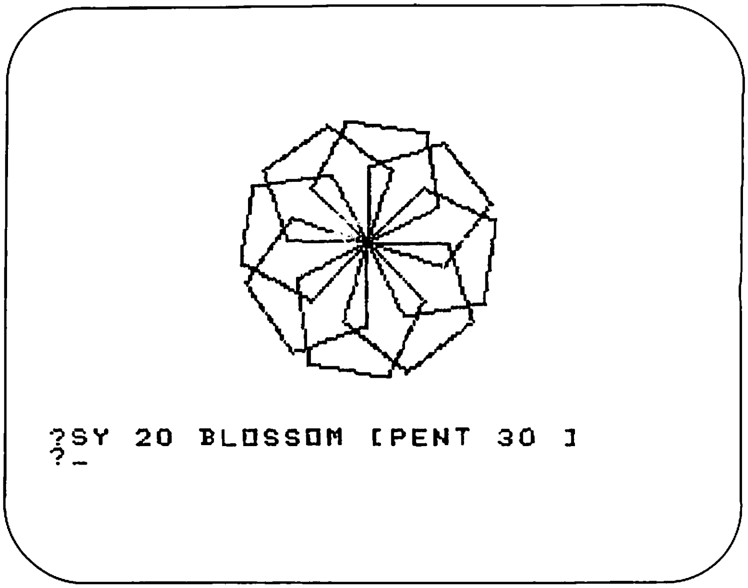


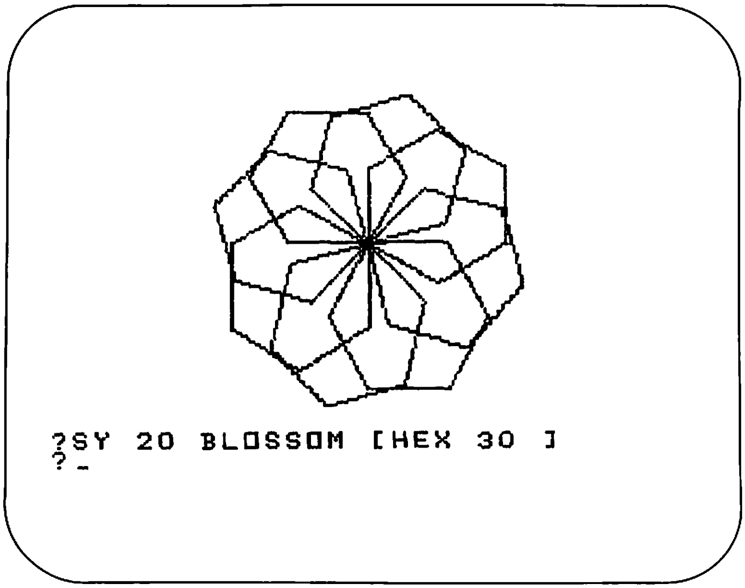
BLOSSOM makes nice patterns with pentagons and hexagons, as you can see by entering

```
CS
SY 20 BLOSSOM [PENT 30]
```

and

```
CS
SY 20 BLOSSOM [HEX 30]
```





POLY and the Single Procedure

Because polygons are such useful building blocks, it would be a great benefit to have a single procedure that would let us create regular polygons with any number of sides. Before we create such a procedure, we should find out what properties are shared by all polygon procedures and see if this helps us design a general procedure.

We can start by looking at the commands that draw triangles, squares, pentagons, and hexagons:

```
REPEAT 3 [FD :SIZE RT 120]
REPEAT 4 [FD :SIZE RT 90]
REPEAT 5 [FD :SIZE RT 72]
REPEAT 6 [FD :SIZE RT 60]
```

These commands all appear pretty similar, except for two numbers. The first number is simply the number of sides in the polygon. The second number is the angle through which the turtle turns before drawing the next side.

As the number of sides in a polygon increases, the turning angle decreases. Let's examine the *total* angle turned by the turtle while drawing each polygon. For the triangle, this number is 3×120, or 360 degrees. For a square, it is 4×90, or 360 degrees. You can easily prove to yourself that this result is also true for pentagons and hexagons. In fact, any closed turtle path that doesn't cross over itself has a total turning angle of 360 degrees. (For the mathematically inclined reader, this is called the *Turtle Total Trip Theorem*. For the nonmathematically inclined reader, feel free simply to enjoy the result without feeling a need to understand its derivation.)

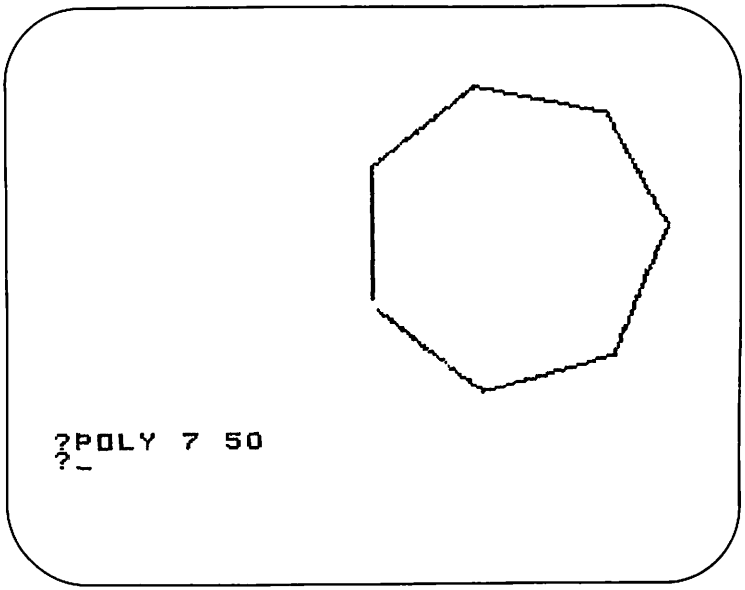
The point of this exercise is that we now know that the turning angle needed for any regular polygon is given by 360 divided by the number of sides. The Logo symbol for division is the solidus (/). See if you can determine that the following procedure, POLY, will let us create any regular polygon. Enter

```
TO POLY :SIDES :SIZE
REPEAT :SIDES [FD :SIZE RT (360 / :SIDES)]
END
```

(We used parentheses to group the calculation 360 / :SIDES as a reminder that this should be done before the turtle turns to the right.)

How does POLY work? Try POLY 4 30 to see if you get a square. So far, so good. Next, enter

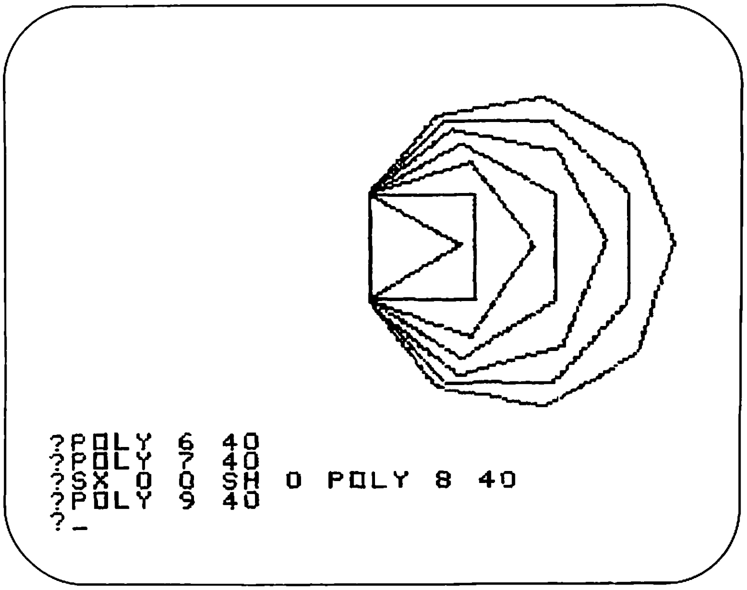
```
CS
POLY 7 50
```



This figure of a heptagon has a small problem: it isn't quite closed. The reason for this error has to do with the way TI Logo performs division. TI Logo can only work with whole numbers (integers). As a result, when we have Logo divide 360 by 7, the result it gets is 51 instead of 51-3/7. This error in division keeps the turtle from turning the correct amount, and so the heptagon isn't quite closed.

To see a collection of polygons, enter

```
CS
POLY 3 40
POLY 4 40
POLY 5 40
POLY 6 40
POLY 7 40
SXY 0 0 SH 0 POLY 8 40
POLY 9 40
```



Procedures such as POLY are quite valuable because they are compact and yet let us create a large number of different figures with very simple commands. Compactness is important because the memory of your TI computer is fixed in size; there is a limit to the number of procedures it can hold. The procedure POLY is the same size as SQUARE, yet it is far more powerful since it lets us create many different polygons. If we erase TRI, SQUARE, PENT, and HEX, we will be able to have more procedures in our workspace without giving up our ability to create triangles, squares, pentagons, or hexagons.

Of course, POLY doesn't let us make all polygons. For polygons with unequal sides or unequal angles, we will have to create new procedures. Let's look at one example.

A Pentagonal Tile

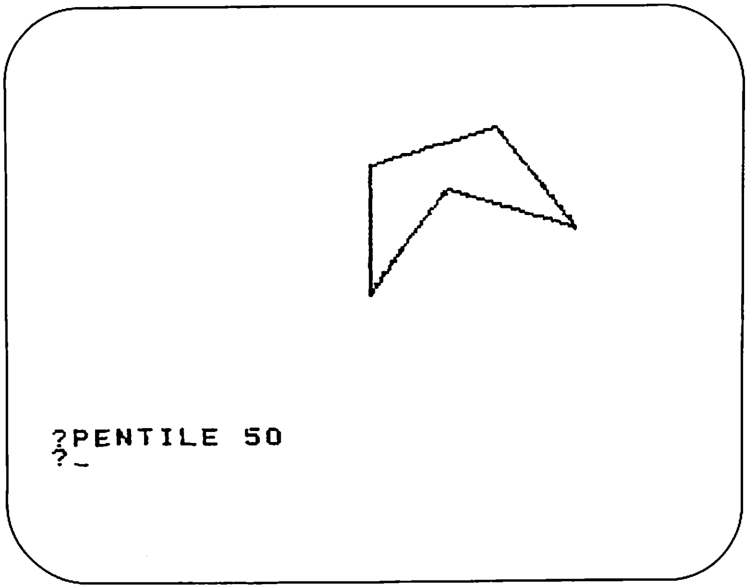
Many different polygons can be combined to create pictures. Sometimes the pictures are representational (such as our use of a square and triangle to make a house), or they can be abstract (such as some of the BLOSSOM patterns). It is interesting to find polygons that can be arranged with other copies of themselves to generate hundreds of different patterns. One such polygon is generated by the PENTILE procedure:

```
TO PENTILE :SIZE
REPEAT 2 [FD :SIZE RT 72]
FD :SIZE RT 144
FD :SIZE LT 72
FD :SIZE RT 144
END
```

(Note that if you add up the angles turned to the right and subtract the angle turned to the left, the result is 360 degrees.)

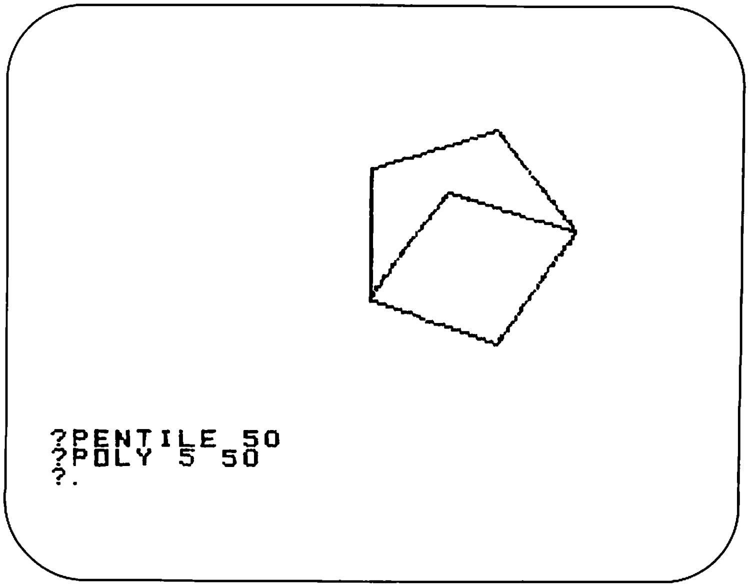
To see a picture of this tile, enter

```
CS
PENTILE 50
```



This figure is a pentagon because it has five sides, but it looks very different from the pentagon created by POLY. To see the relation between the two, enter

POLY 5 50



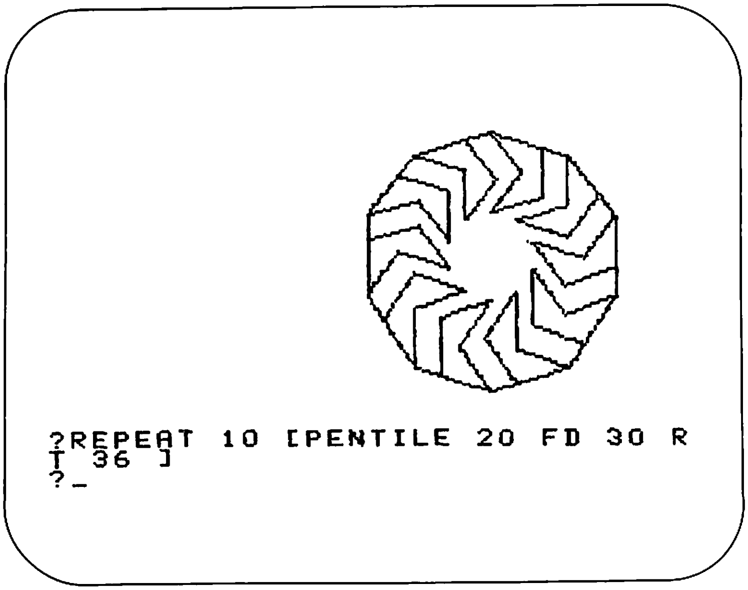
The only difference is that PENTILE was made by snapping in one of the corners of our regular pentagon.

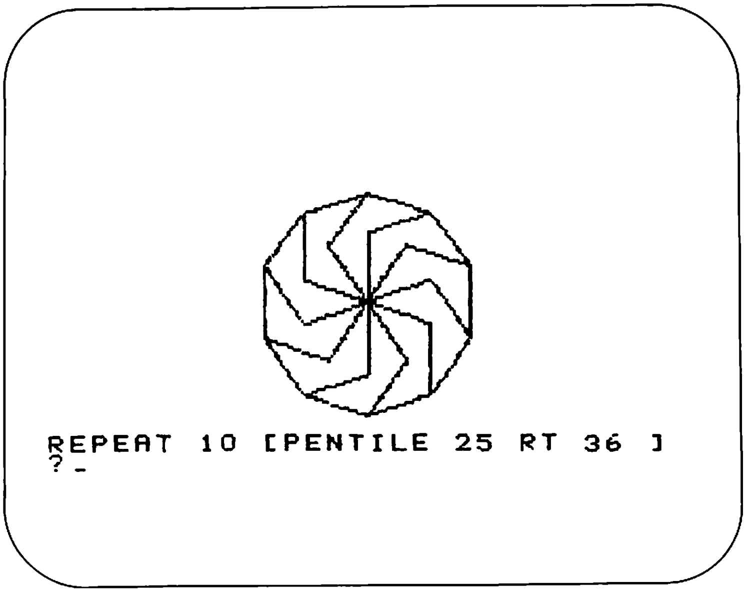
PENTILE polygons can be arranged in many interesting patterns. You may even want to cut some out of cardboard so that you can experiment when you are away from the computer. (A beautiful wooden set of tiles based on this figure is availale in toy stores under the tradename PENTALBI.) The following two figures show just a few of the many patterns you can create with this polygon. These figures were created with the commands

```
CS
REPEAT 10 [PENTILE 20 FD 30 RT 36]
```

and

```
CS
REPEAT 10 [PENTILE 25 RT 36]
```





Some Projects with Polygons

You may find that polygons are among your favorite building blocks. You should be able to create several pictures using polygons. Using PENTILE and POLY, for example, make a simple BOAT procedure. Can you next create a regatta? Use POLY to make a fanciful steam engine or other mechanical contraption.

After you have tried a few such pictures, use POLY (and PENTILE) to create some abstract pictures. Use many different pen colors. When you have completed your pictures, change the background colors to see which ones you like best. As always, save your pictures for later viewing.

VI. Circles, Arcs, and Stars

The preceding chapter provided a simple way to generate regular polygons. Although these polygons are very valuable building blocks for graphic creations, they are insufficient for the creation of many figures. This chapter will show ways to create other geometric shapes of great power—circles, arcs, and stars.

Before proceeding with this chapter, you should erase all the procedures in your workspace except POLY and BLOSSOM.

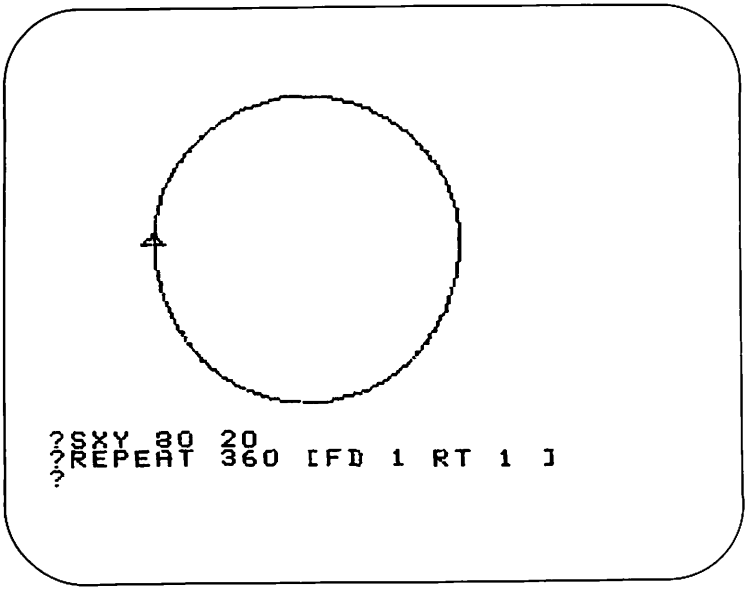
Circles

Circles are among the most powerful geometric symbols. They convey perfection, wholeness, and a sense of peace that does not exist in any other simple figure.

To have the turtle generate a circle, we must find a way to send it on a circular path. This may seem challenging, since the turtle can either turn or move in a straight line but not both simultaneously. We can create a useful approximation to a circle, however, by having the turtle take a small step and then turn a slight amount before moving again. As this process is repeated, the turtle will move in a circular path. The smallest step the turtle can take is one unit, and the smallest turning angle is one degree, so we should try this combination to see what it generates. From the turtle total trip theorem, we know that we will need 360 such steps to complete the figure.

To see our circle, enter

```
TELL TURTLE CS
SXY -80 20
REPEAT 360 [FD 1 RT 1]
```



Technically, this figure is not a circle—it is a 360-sided polygon; but the granularity of the display screen's resolution wouldn't let us generate any more detail, even if we could increase the number of sides. (*Note:* If your circle looks elliptical, your display's aspect ratio is not set properly. As was mentioned earlier, you may be able to fix this if your TV set has a vertical size control.)

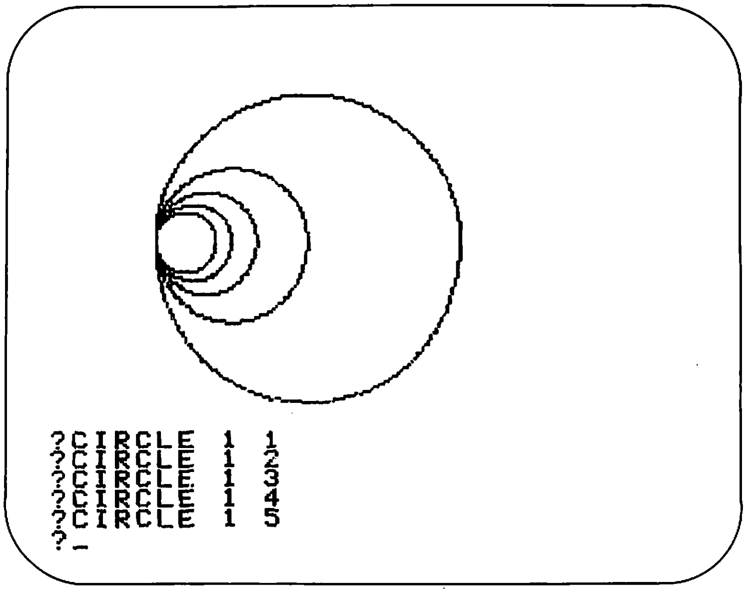
Now that we have created a circle, we should find out how to create circles of different sizes. If we increase the size of each step the turtle takes, we will get a larger circle; if we increase the angle turned at the end of each step, we will get a smaller circle. The following procedure lets us experiment with different circles:

```
TO CIRCLE :SIZE :ANGLE
REPEAT (360 / :ANGLE) [FD :SIZE RT :ANGLE]
END
```

First, let's experiment with different values of ANGLE by entering

```
CS HT
SXY -80 20
CIRCLE 1 1
CIRCLE 1 2
CIRCLE 1 3
CIRCLE 1 4
CIRCLE 1 5
```

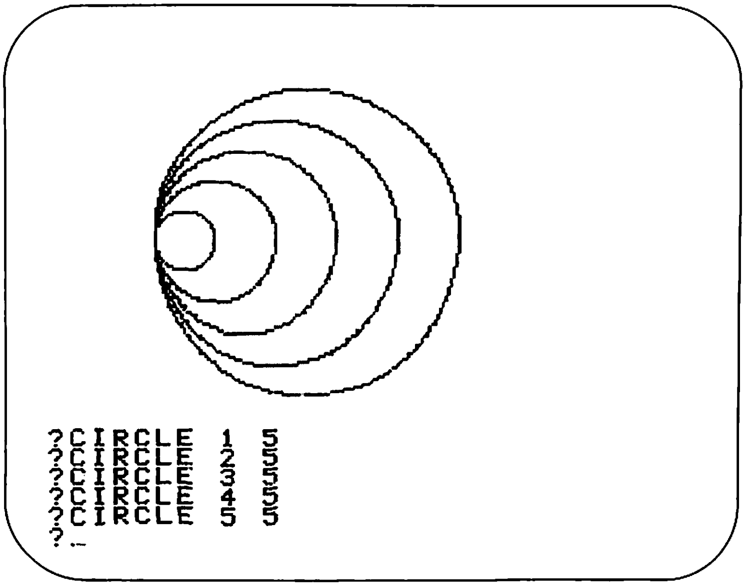
This gives us ever smaller circles.



Next, let's experiment with different values of SIZE by entering

```
CS HT
SXY -80 20
CIRCLE 1 5
CIRCLE 2 5
CIRCLE 3 5
CIRCLE 4 5
CIRCLE 5 5
```

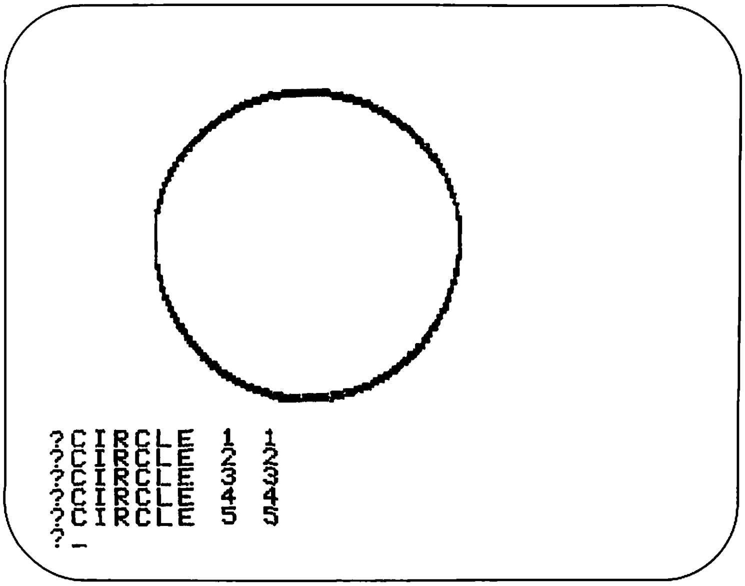
As you see, this gives us ever larger circles.



Next, let's change SIZE and ANGLE at the same time to see what that gives us. Enter

```
CS HT
SXY -80 20
CIRCLE 1 1
CIRCLE 2 2
CIRCLE 3 3
CIRCLE 4 4
CIRCLE 5 5
```

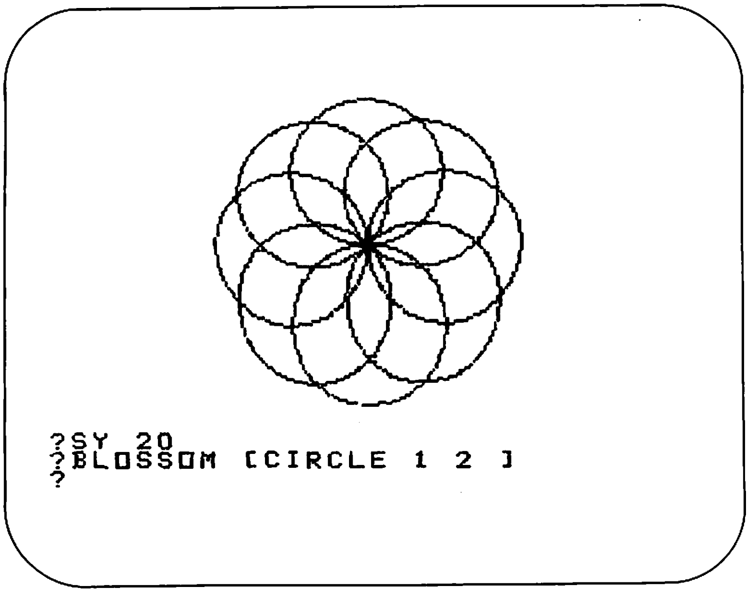
All these circles are nearly the same size. The differences in their shapes occur because we are really drawing polygons with different numbers of sides.



As you experiment with CIRCLE, you will find values of SIZE and ANGLE that let you create any size circle you want.

Next, try

```
CS
SY 20
BLOSSOM [CIRCLE 1 2]
```



Now that we know how to make circles, we should learn how to make parts of circles.

Arcs

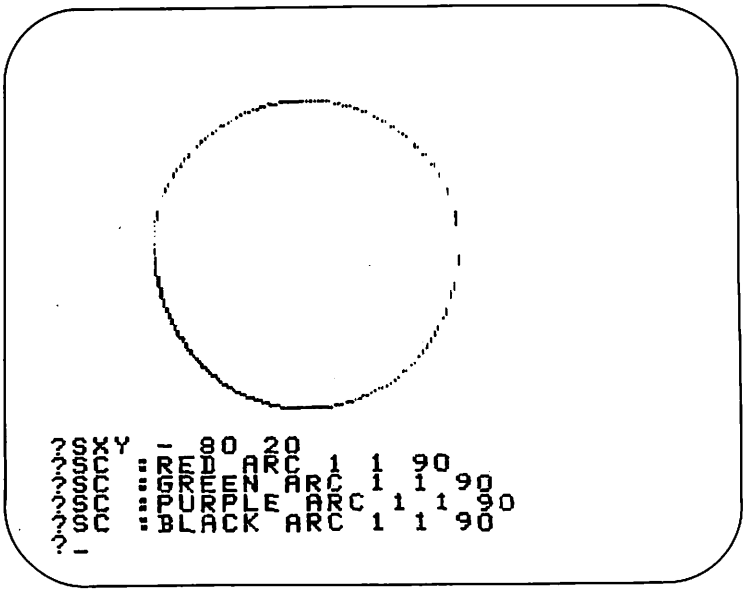
If we want to draw only part of a circle, we must modify the CIRCLE procedure so that it can be stopped before we have turned 360 degrees. If we do travel 360 degrees, the turtle will trace a closed path.

Let's experiment by using the following procedure:

```
TO ARC :SIZE :ANGLE :AMOUNT
REPEAT (:AMOUNT / :ANGLE) [FD :SIZE RT :ANGLE]
END
```

We can test this ARC procedure by creating a circle from four 90-degree arcs. Enter

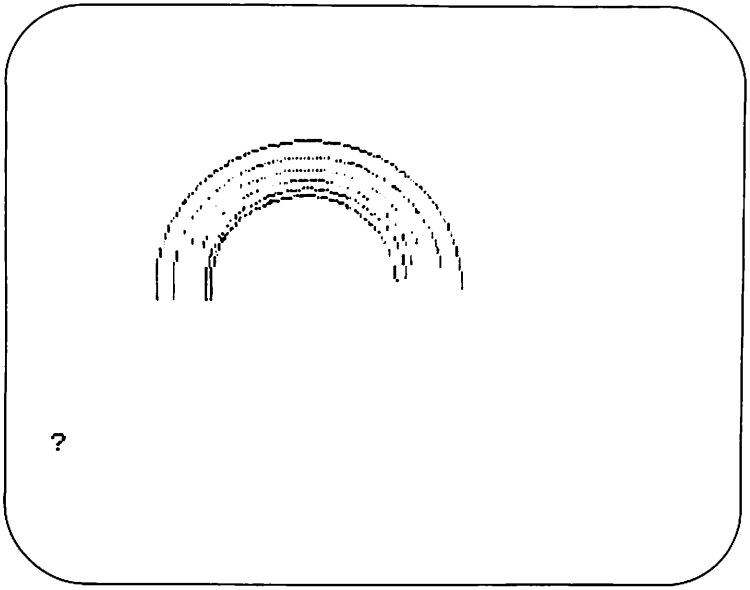
```
CS
SXY -80 20
SC :RED ARC 1 1 90
SC :GREEN ARC 1 1 90
SC :PURPLE ARC 1 1 90
SC :BLACK ARC 1 1 90
```



By adjusting the values for SIZE, ANGLE, and AMOUNT, you will be able to create arcs that are long and graceful as well as arcs that are small and tightly turned.

As an experiment, create a rainbow using arcs. Here is one procedure that will do it for us:

```
TO RAINBOW
TELL TURTLE CS
SC :RED SXY -80 0 SH 0 ARC 8 8 185
SC :ORANGE SXY -74 0 SH 0 ARC 8 9 185
SC :YELLOW SXY -70 0 SH 0 ARC 8 10 185
SC :GREEN SXY -66 0 SH 0 ARC 8 11 185
SC :BLUE SXY -62 0 SH 0 ARC 8 12 185
SC :PURPLE SXY -60 0 SH 0 ARC 8 13 185
END
```

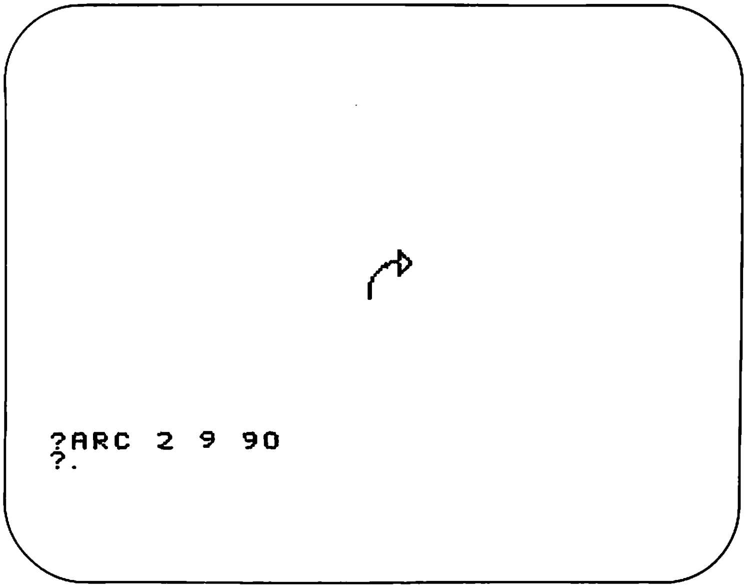


A Flower

Using the procedures we have created, we can draw a pretty picture of a flower. Since we know how to use BLOSSOM for the flower itself, and we know that ARC can be used to create the stem, let's start with a procedure for creating a leaf.

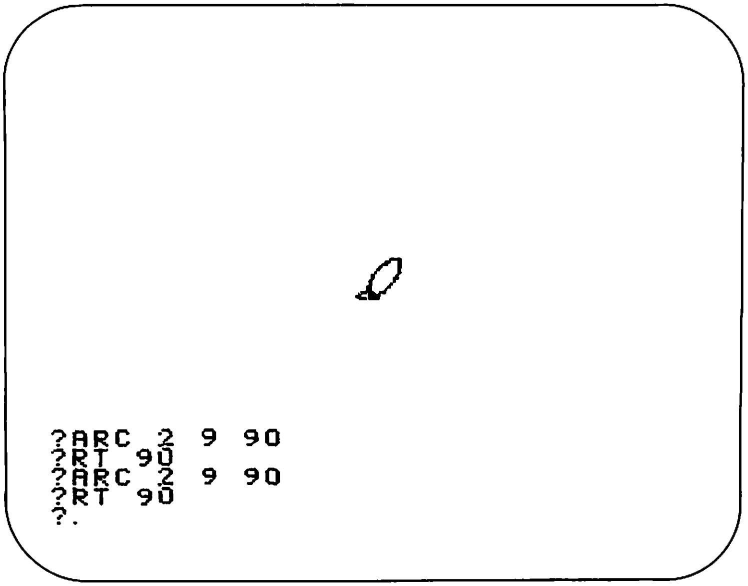
A leaf can be made from two small arcs. As an example, enter

```
CS ST
ARC 2 9 90
```



This draws half of the leaf. To draw the other half, we must first turn the turtle, or we will draw a semicircle instead. Since the turtle needs to turn by 360 degrees overall, and since the two arcs for the leaf sides account for 180 degrees, we must turn an additional 180 degrees. We want to have the turtle back at its starting position when we are done, so we should turn the turtle by 90 degrees after each use of ARC. To try this, enter

```
RT 90
ARC 2 9 90
RT 90
```



This generates a nice leaf for us. We can now define the LEAF procedure as follows:

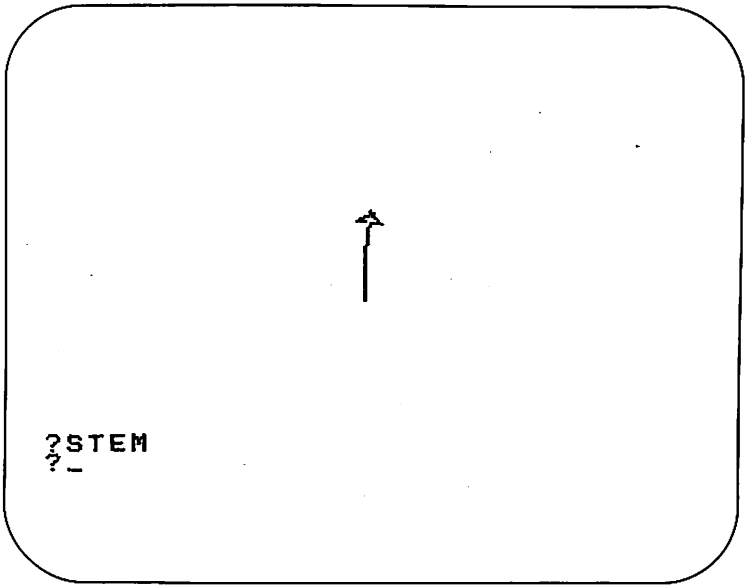
```
TO LEAF
REPEAT 2 [ARC 2 9 90 RT 90]
END
```

Next, we need a procedure that draws a portion of the stem. To draw the stem, let's use a gradual arc. For the STEM procedure, enter

```
TO STEM
ARC 3 1 10
END
```

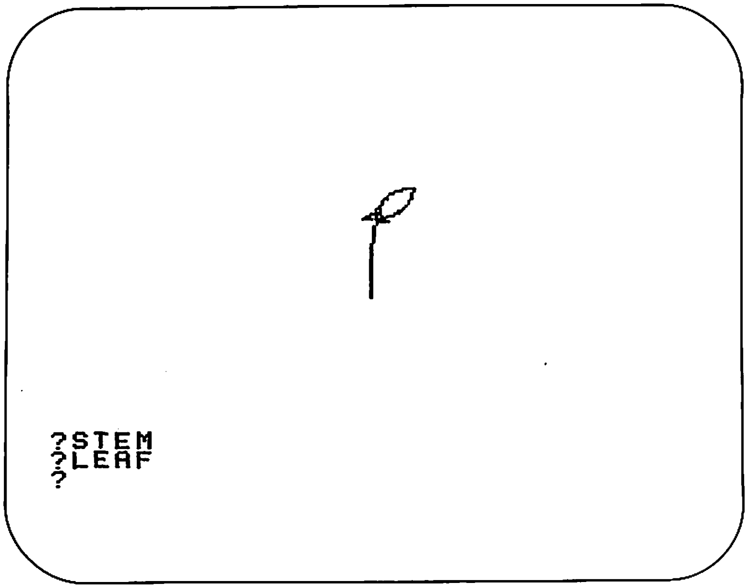
Now, to draw our flower, let's start by drawing the stem. Enter

```
CS
SC :BLACK
STEM
```



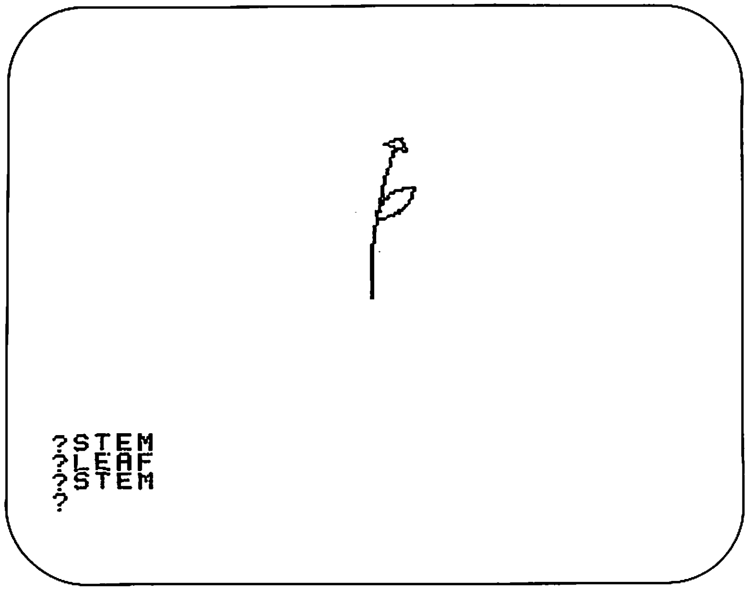
Next, let's draw a leaf, by entering

```
LEAF
```



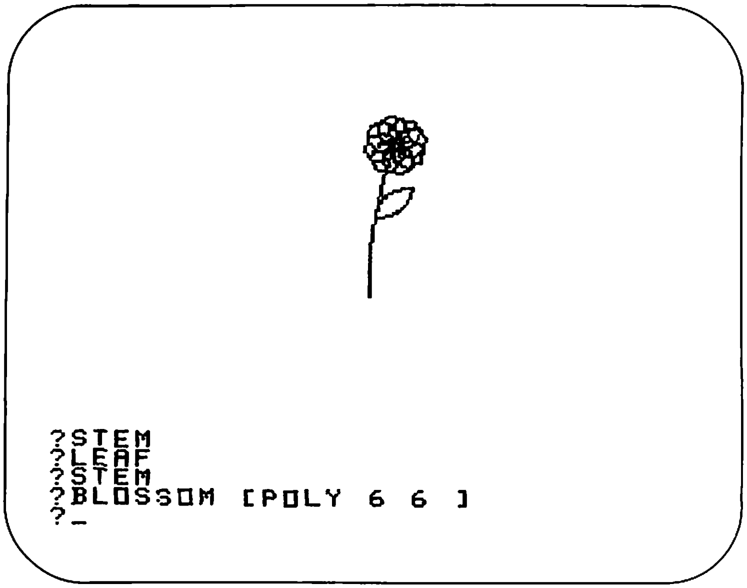
We can extend the stem some more by entering

STEM



Now we can make the blossom of our flower by entering

```
BLOSSOM [POLY 6 6]
```



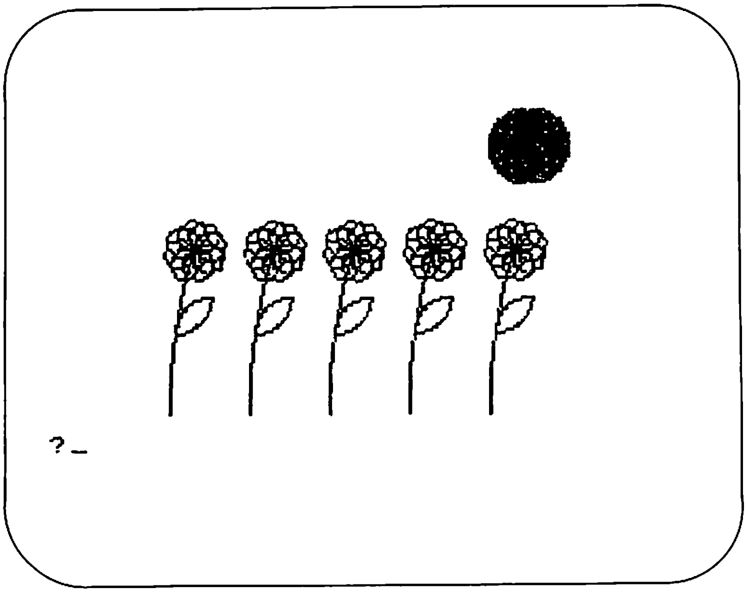
Now that all the pieces work together properly, we can create a FLOWER procedure that lets us draw flowers of different colors:

```
TO FLOWER :COLOR
HT
SC :GREEN
STEM
LEAF
SC :COLOR
STEM
BLOSSOM [POLY 6 6]
END
```

When you start using FLOWER, you will notice that the turtle does not end at its beginning position. Keep this in mind as you use this procedure.

Let's make a picture using FLOWER. The following PICTURE procedure will draw a small row of flowers and put a yellow sun in the sky:

```
TO PICTURE
TELL TURTLE CS HT
SXY -75 (-40) SH 0 FLOWER :RED
SXY -45 (-40) SH 0 FLOWER :ORANGE
SXY -15 (-40) SH 0 FLOWER :BLUE
SXY 15 (-40) SH 0 FLOWER :RUST
SXY 45 (-40) SH 0 FLOWER :YELLOW
SXY 60 60 SC :YELLOW
REPEAT 90 [FD 15 BK 15 RT 4]
END
```



The last command you typed gave us a yellow sun. If we had wanted a night scene, we could have put stars in the sky instead.

Stars

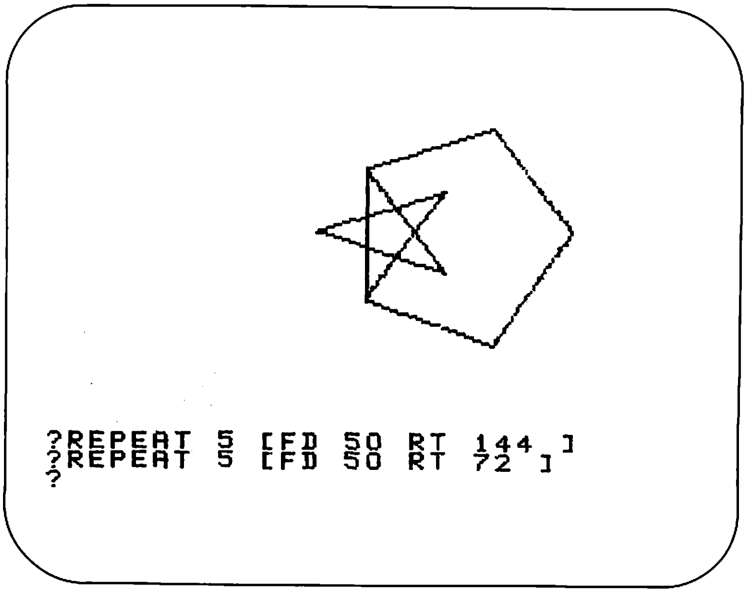
We already know that a five-pointed star can be drawn with a command such as

```
REPEAT 5 [FD 50 RT 144]
```

In this section, we will explore this and other stars, and we will learn how stars are related to the simple polygons we created in the last chapter.

For starters, let's compare the five-pointed star with a pentagon. Enter

```
CS
REPEAT 5 [FD 50 RT 144]
REPEAT 5 [FD 50 RT 72]
```



The only difference between the commands for these two figures is the turning angle. You might have noticed that 144 is twice 72. This has an interesting result, suggesting a relationship between stars and simple polygons. As we will see, however, not all polygons can be turned into stars by doubling the turning angle. Also, some polygons can be turned into different stars by doubling or tripling the turning angle or increasing it by other multipliers.

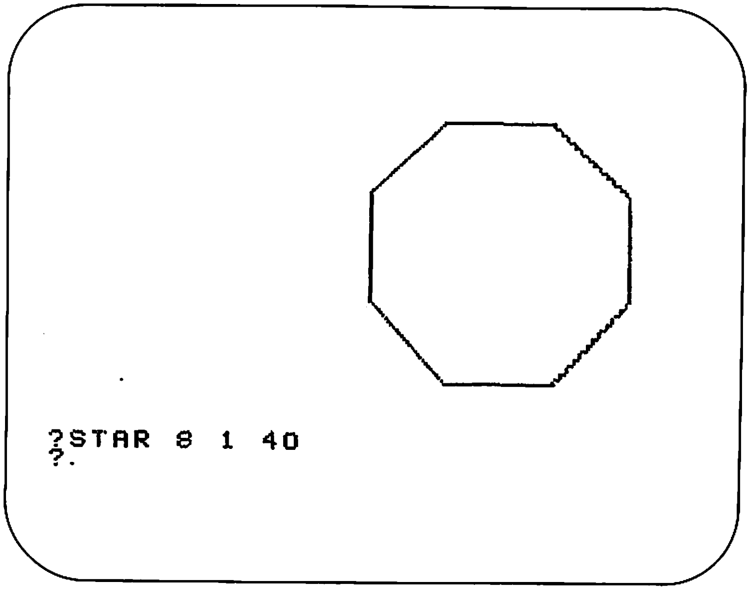
We can create a generalized star procedure that is similar in structure to POLY by entering

```
TO STAR :SIDES :MULT :SIZE
REPEAT :SIDES [FD :SIZE RT (:MULT * 360 / :SIDES)]
END
```

Note that Logo uses the asterisk (*) as the symbol for multiplication.

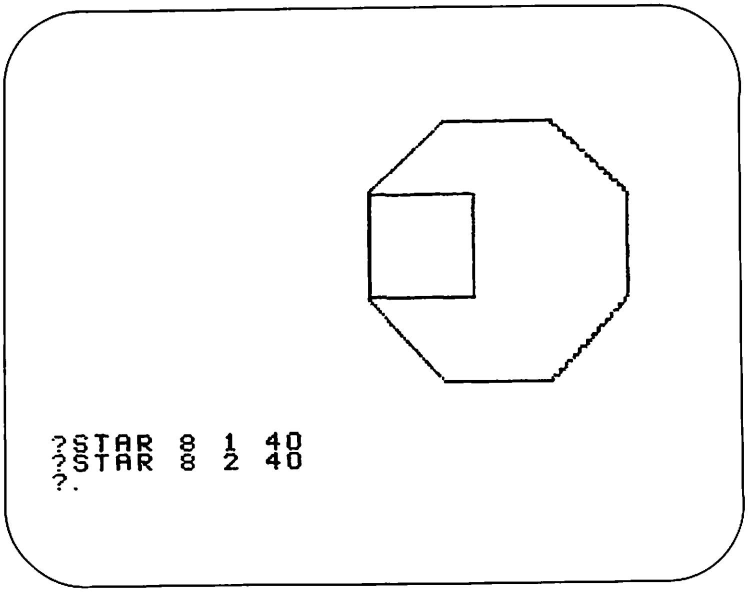
Now that you have the STAR procedure, let's test it. Enter

```
CS
STAR 8 1 40
```



This generated an octagon—the expected result for a multiplier of one. Next, enter

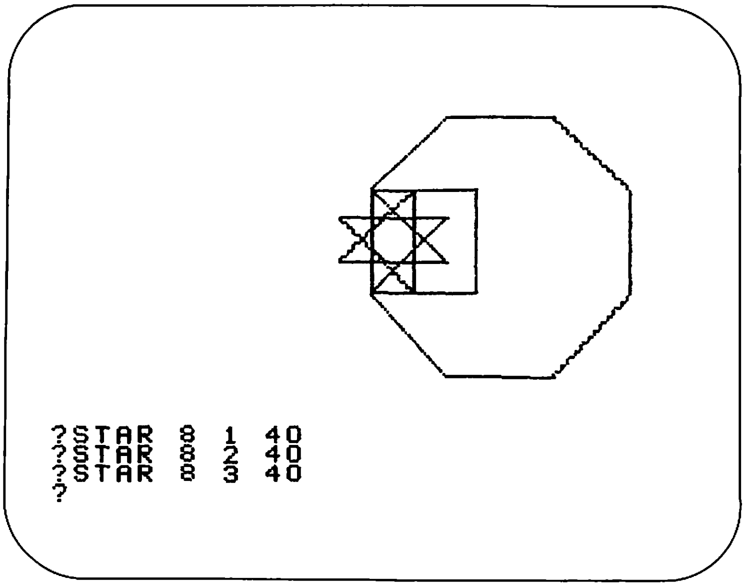
STAR 8 2 40



This added a square to the figure! The square results from a turning angle of 2 × 360/8, or 90 degrees. If we now enter

STAR 8 3 40

we finally get an eight-pointed star!



You should experiment with different values for SIDES and MULT to see which combinations give stars and which do not.

Some Projects

Now that circles, arcs, and stars have been added to your graphics tool kit, you should be ready to create some very interesting pictures. As a starter project, try to modify PICTURE to generate a night scene instead of the day scene we created. To do this, change the background to black and replace the sun with small stars in the sky. See how many stars you can add before the computer "runs out of ink." Remember that with a black background, you won't be able to read any of the computer's messages. When you are done with your picture, return the background to cyan.

Try using different star patterns with BLOSSOM. Are these more interesting than the patterns BLOSSOM creates with simple polygons?

Can you create a diamond that fits inside a circle so that its corners just touch the circle's boundary?

Using your knowledge of lines, polygons, circles, and arcs, can you create the signs of the zodiac?

Using the counter technique described in Chapter 4, create a growing star on the screen. As the figure is being drawn, do you sense the illusion of motion?

As always, save your artwork for later viewing. Because Logo's memory can become filled easily, you should save each picture in a separate file. This will become more important as we create more complex pictures.

VII. Squirals and Spirals

Thus far, we have explored graphic procedures that produce static images. When we look at their finished forms, there is no clue to the process that created them. In this chapter, we will explore some figures that are formed by a sequential growth process. As with polygons, we will be dealing with a set of commands that are repeated to form the object. By increasing the size of the lines drawn with each repetition of the command sequence, an object is caused to grow on the display screen.

Squirals

Living things sometimes leave traces of their growth patterns that can be studied without watching the object grow. Seasonal cycles, for example, produce a series of concentric rings in trees. By counting the rings, we can deduce the age of a tree. A more common growth pattern, found in both plants and animals, is the spiral. The effect of spirals on the eye is so strong that they almost appear to be in motion.

The first spiral figures we will explore are made with straight lines. These spirals are called *squirals* (from *square spirals*). Before entering the procedures used in this chapter, you should clear your workspace.

To start, let's examine a spiral made from straight lines and square (90-degree) corners. To draw a square, we draw the same length line after each turn of 90 degrees. To draw a square spiral, we need to increase the length of each side over its previous value. The following procedure will allow us to experiment with squirals containing various turning angles. The increment by which each side grows is chosen to be 2, although you may want to change this value as you experiment.

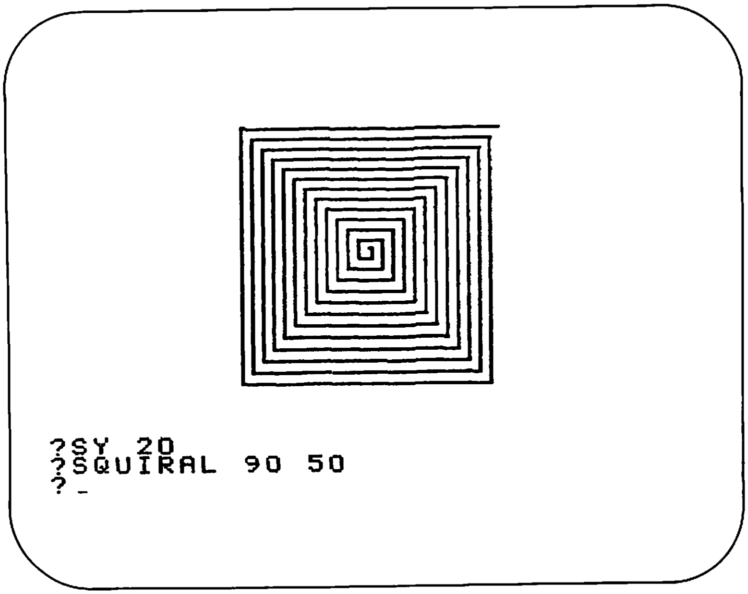
To create the SQUIRAL procedure, enter

```
TO SQUIRAL :ANGLE :STEPS
MAKE "SIDE 0
REPEAT :STEPS [FD :SIDE RT :ANGLE MAKE "SIDE
   :SIDE + 2]
END
```

Next, enter

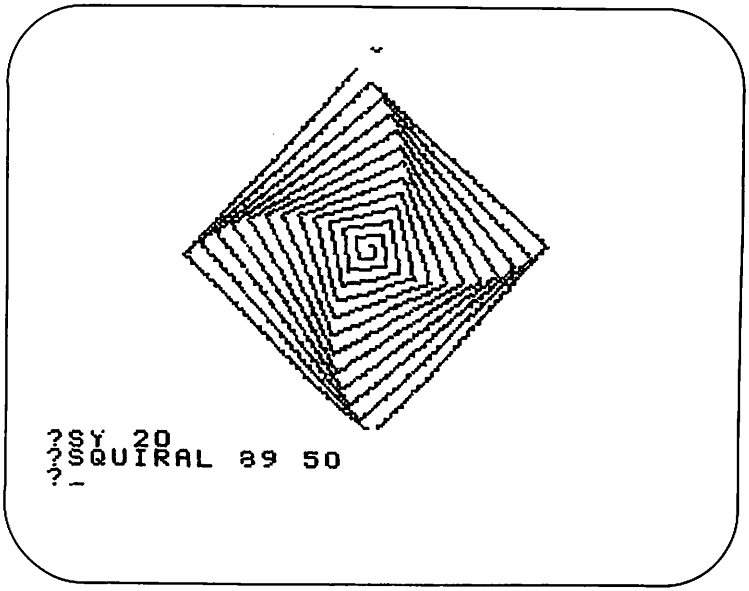
```
TELL TURTLE CS HT
SY 20
SQUIRAL 90 50
```

This will generate a squiral pattern on the display screen.



On examination, we can see the close relationship of this figure to the square. Next, suppose we were to use an angle close to 90 degrees—89 degrees, for example. Before trying this angle, try to visualize the result in your mind's eye. Will the change be small or large? Once you have made your decision, enter

```
CS
SY 20
SQUIRAL 89 50
```



This figure looks very different from the figure generated by SQUIRAL 90 50. Why is this?

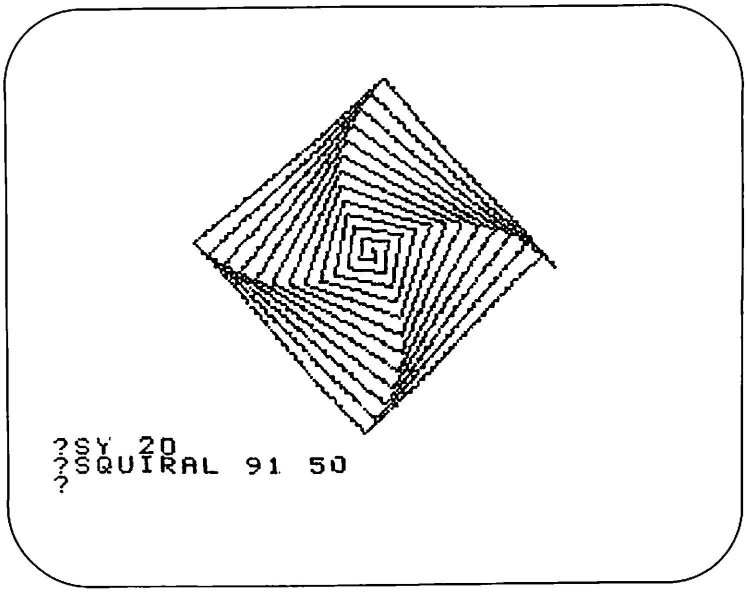
As the figure for SQUIRAL 89 50 is drawn, each turning angle differs from that for a square by only 1 degree, but the impact of this difference is cumulative so far as the overall figure is concerned. By the time the turtle has made one circuit around the path, the difference is 4 degrees. By the time the procedure stops, the turning angle is 100 degrees in variance from its value for the truly square squiral.

One of the attractions of this particular figure is the four arching branches twisting to the left. These branches are formed as an interference pattern of corners that bump into each other because of the angular mismatch (compared to a square). Such interferences, called moiré patterns, are quite common. You can see moiré patterns by holding two window screens together and rotating one of them slightly with respect to the other. When the screens are adjusted so that there is no interference pattern, the screens are perfectly aligned.

We can see a similar effect for squirals as well. As an example, if we enter

```
CS
SY 20
SQUIRAL 91 50
```

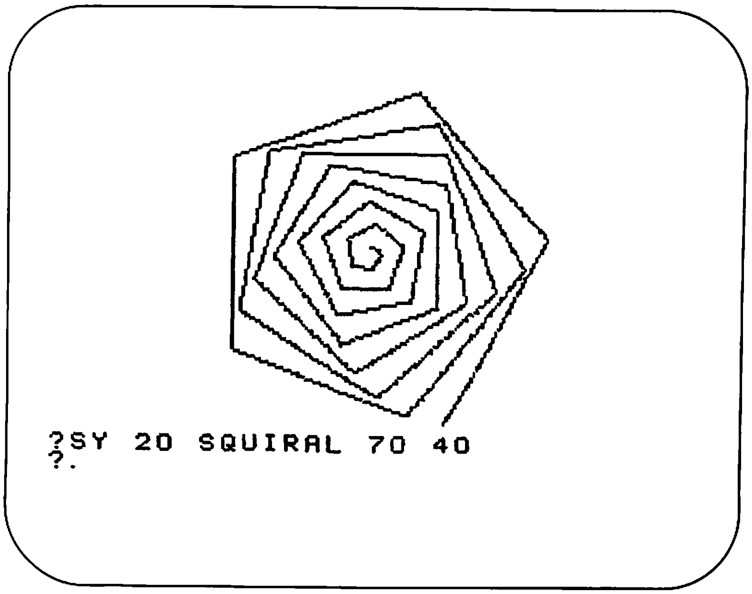
we will generate a squiral pattern that has arms branching off to the right.

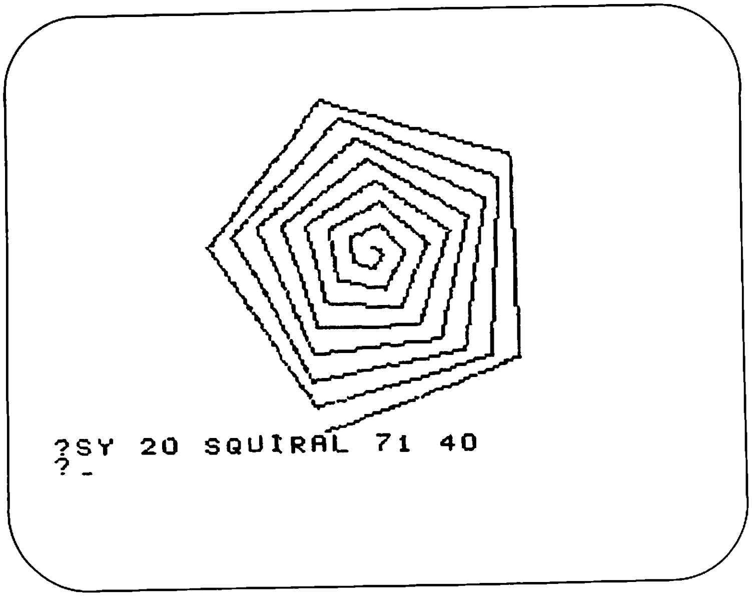


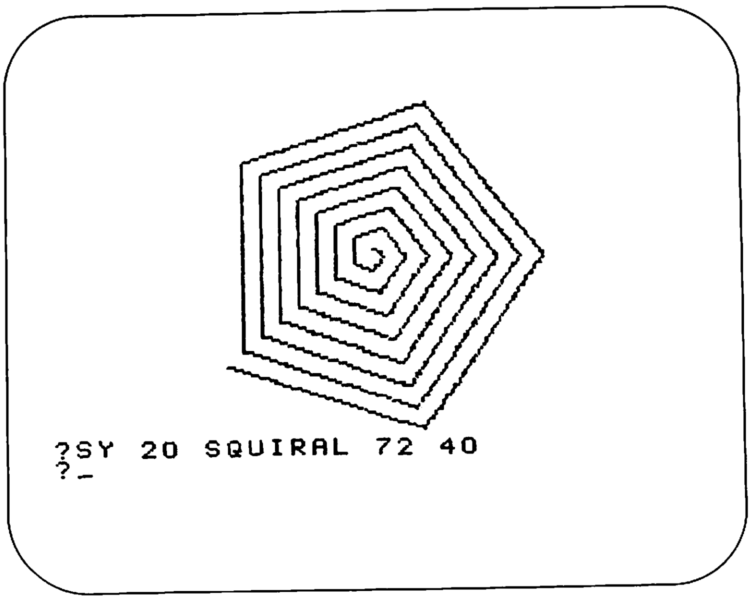
Whenever we see interferences of this sort, it is a clue that we are close to a regular pattern whose alignment of sides creates no interference. If the squirals arch to the left, the angle is too small; if they arch to the right, the angle is too large.

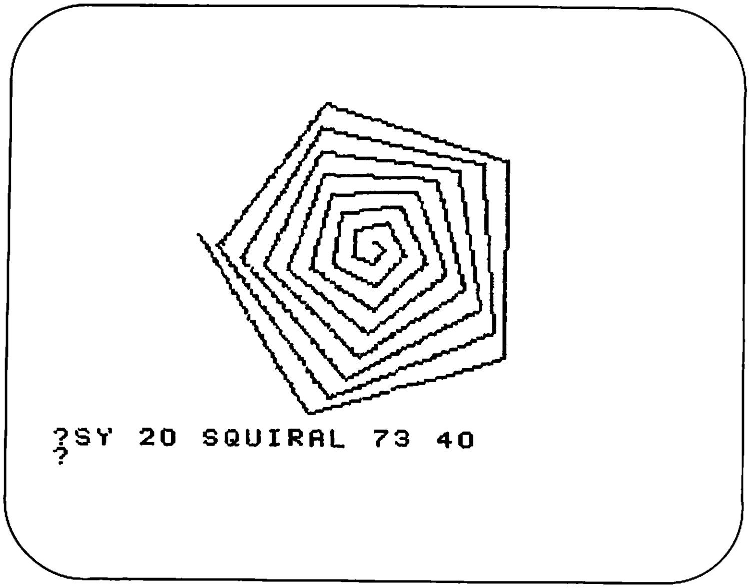
How many squirals are there with no interference patterns? The following figures are examples of squiral patterns based on the pentagon. (You will want to clear the screen before drawing each squiral.)

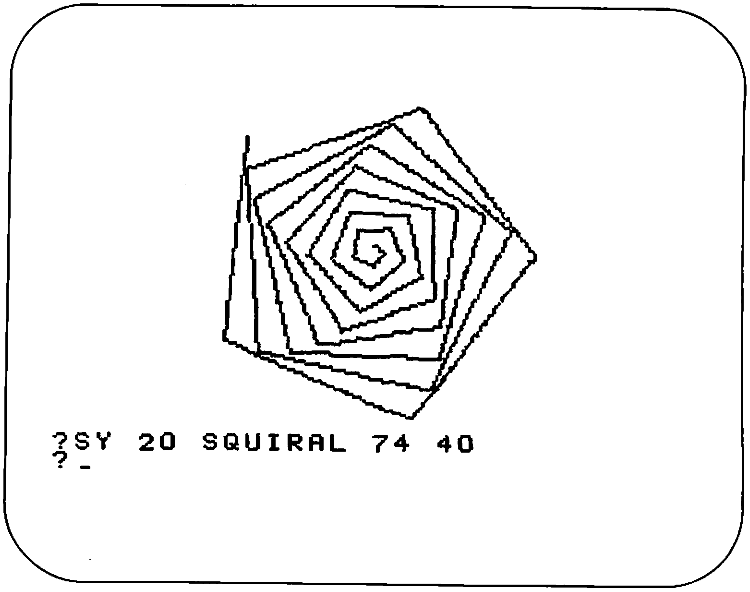
```
SQUIRAL 70 40
SQUIRAL 71 40
SQUIRAL 72 40
SQUIRAL 73 40
SQUIRAL 74 40
```




?SY 20 SQUIRAL 71 40
?_


?SY 20 SQUIRAL 72 40
?_

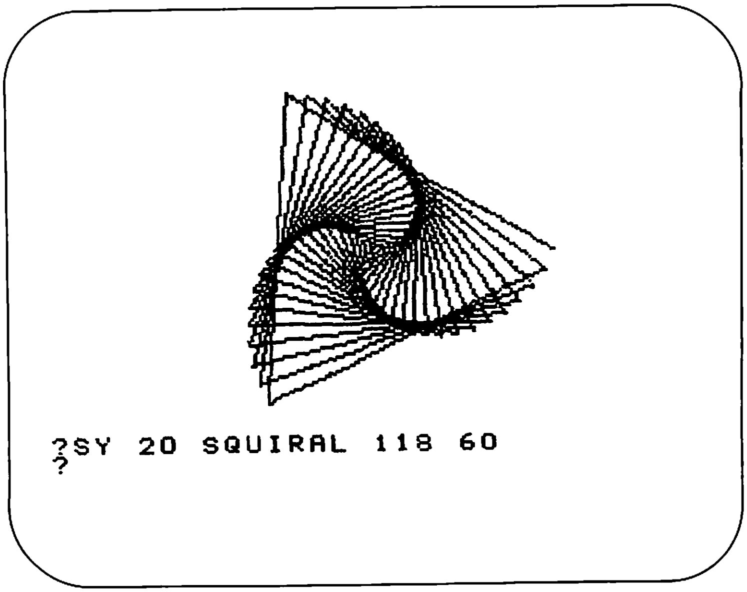

?SY 20 SQUIRAL 73 40
?

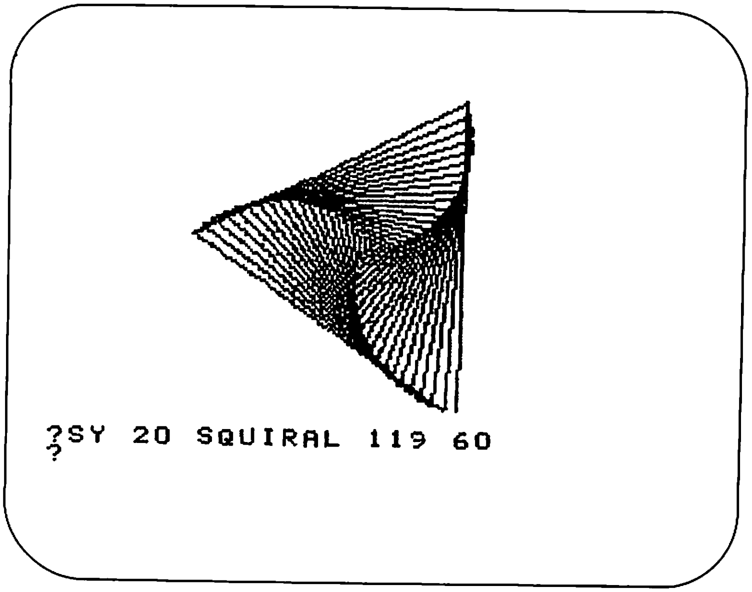

?SY 20 SQUIRAL 74 40
? _

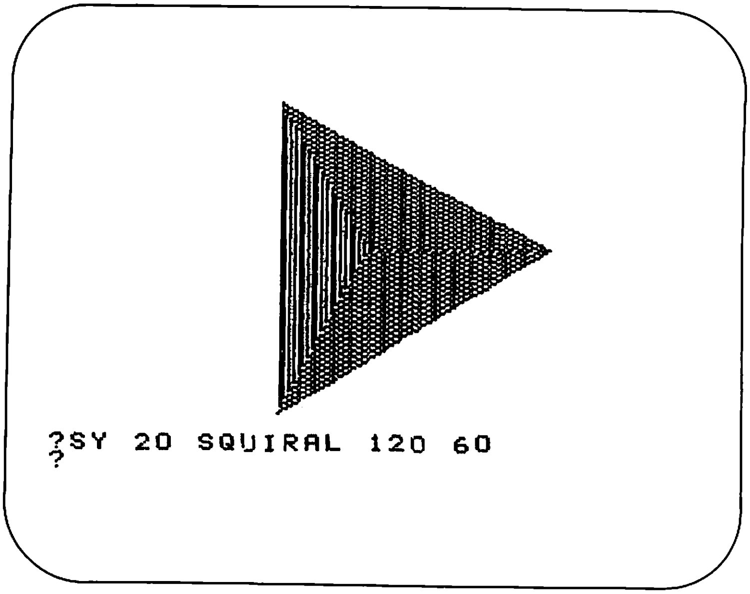
Once again, we can see the spiral arms in all patterns except SQUIRAL 72 40. The 72-degree angle is the angle associated with a regular pentagon. Notice that the curvature of the spiral arms is greater as you move farther away from 72 degrees in either direction.

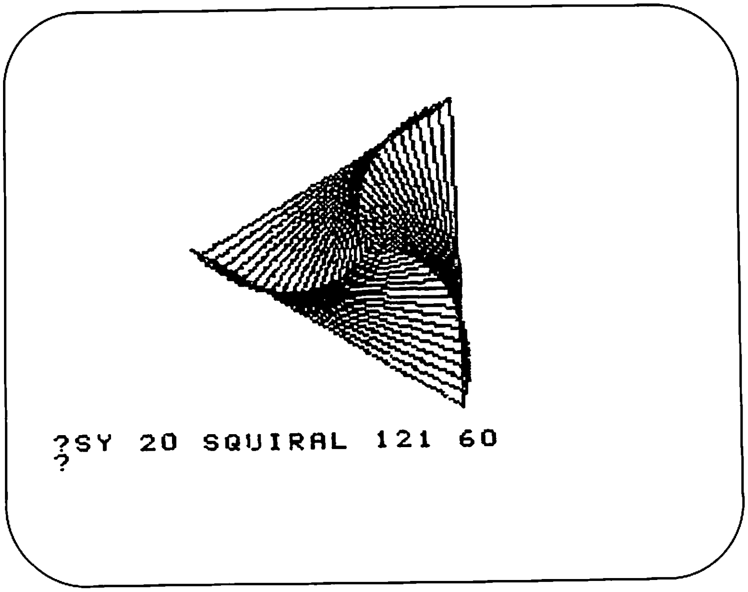
The next set of figures explores squirals in the vicinity of 120 degrees:

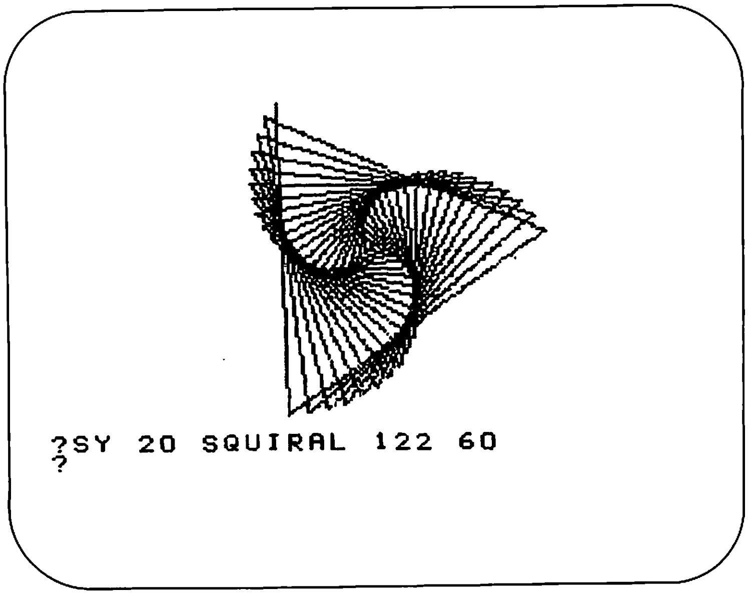
SQUIRAL 118 60
SQUIRAL 119 60
SQUIRAL 120 60
SQUIRAL 121 60
SQUIRAL 122 60




?SY 20 SQUIRAL 119 60
?

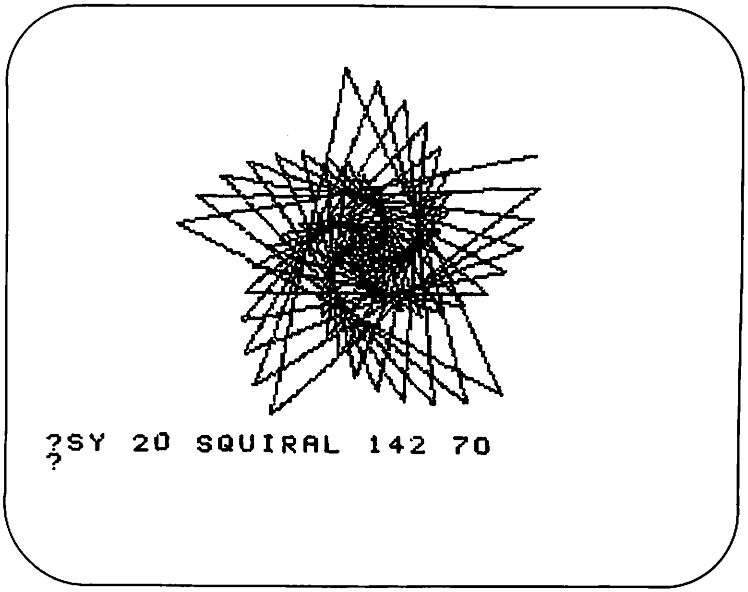

?SY 20 SQUIRAL 120 60
?

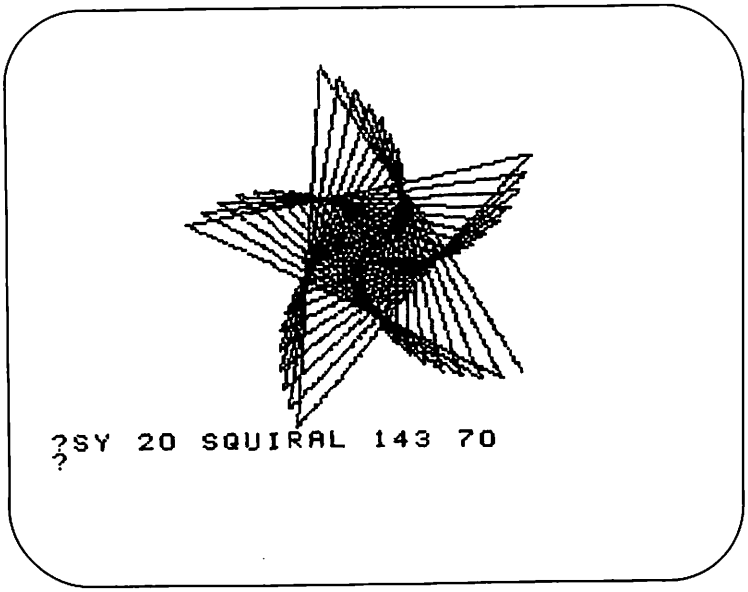

?SY 20 SQUIRAL 121 60
?

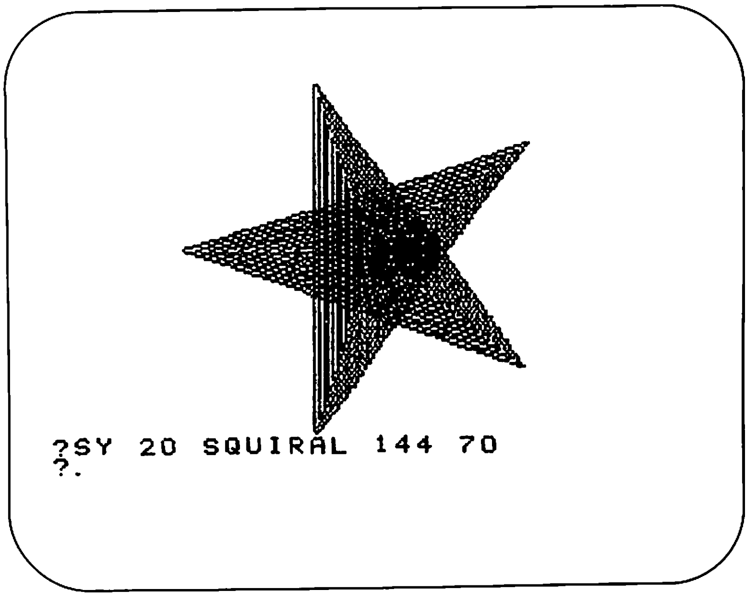

?SY 20 SQUIRAL 122 60
?

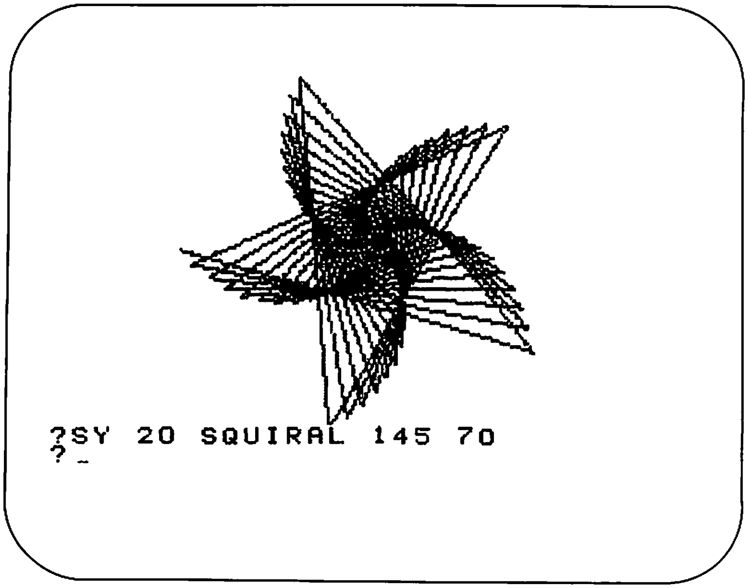
Squiral patterns need not be based on simple polygons. The following attractive figures, for example, occur in the vicinity of 144 degrees:

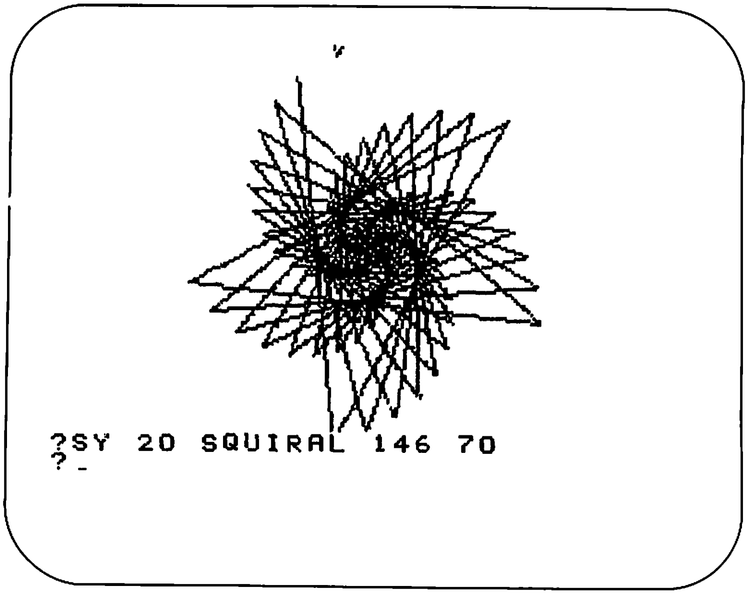
```
SQUIRAL 142 70
SQUIRAL 143 70
SQUIRAL 144 70
SQUIRAL 145 70
SQUIRAL 146 70
```




?SY 20 SQUIRAL 143 70
?


?SY 20 SQUIRAL 144 70
?.


?SY 20 SQUIRAL 145 70
? _


?SY 20 SQUIRAL 146 70
? _

As you experiment with the SQUIRAL procedure, you will find many interesting patterns. Each of these patterns reflects the process of growth by which it was created.

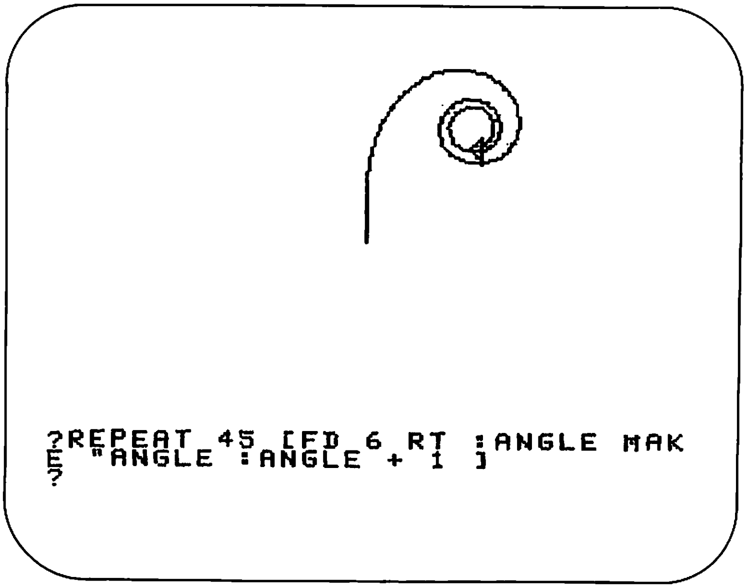
Closed Spirals

In the preceding section, we created squirals by keeping the turning angle fixed and increasing the size of the drawn lines. By reversing this sequence, we can create some interesting spirals. To make a spiral curve, we can draw a series of fixed-length lines and turn the turtle by increasing amounts at the end of each step.

Rather than starting with a spiral procedure, we will create a spiral interactively by using primitive Logo commands. As an example, enter

```
SHOWTURTLE
CS
SY 20
MAKE "ANGLE 0
REPEAT 45 [FD 6 RT :ANGLE MAKE "ANGLE :ANGLE + 1]
```

This set of instructions starts the turtle off on a gentle arc to the right that begins to circle in on itself quickly.



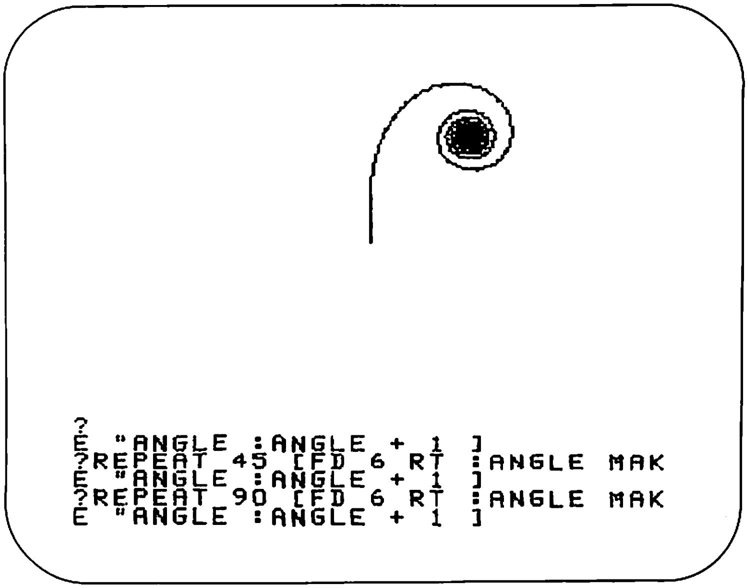
Next, enter

REPEAT 45 [FD 6 RT :ANGLE MAKE "ANGLE :ANGLE + 1]

This continues to tighten the spiral. If we take an additional 90 steps (making 180 in total) by entering

REPEAT 90 [FD 6 RT :ANGLE MAKE "ANGLE :ANGLE + 1]

we will bring the turtle to the center of the circular area that is forming on the screen.

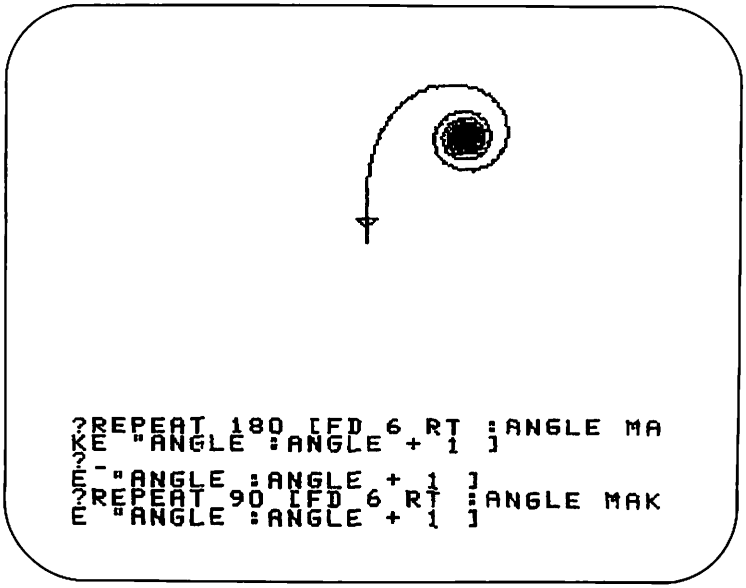


What will happen with angles greater than 180 degrees? When we started, the turtle turned first by 1 degree, then by 2 degrees, and so on. After the first four steps, the turtle had turned by only 10 degrees. Contrast this with the turning that took place at the four steps starting at 89 degrees. After those four steps, the turtle had turned over 360 degrees. By the time we reached 180-degree increments, the turtle was simply moving back and forth over its position.

To see what happens for increments greater than 180 degrees, enter

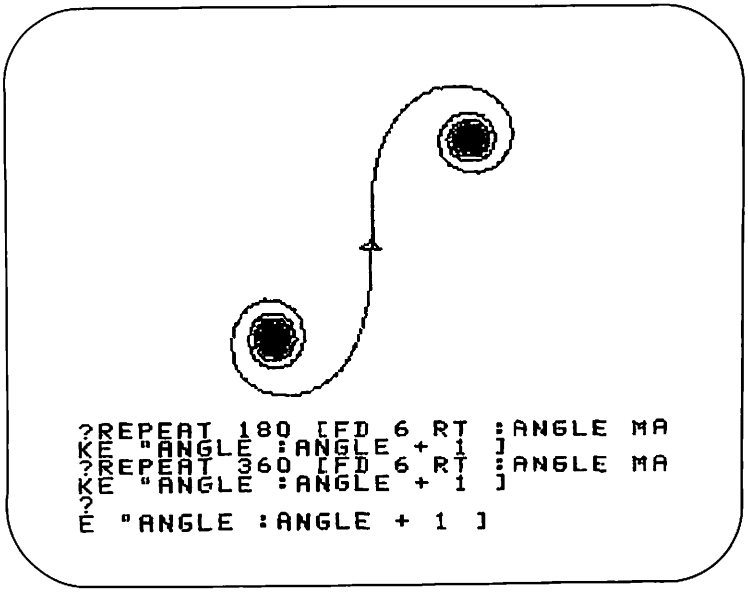
```
REPEAT 180 [FD 6 RT :ANGLE MAKE "ANGLE :ANGLE + 1]
```

At the end of 360 steps, the turtle is back at the origin, having retraced its steps. The turtle is pointing down, however, instead of up.



To complete the figure (and return the turtle to its home position), we need to take an additional 360 steps:

```
REPEAT 360 [FD 6 RT :ANGLE MAKE "ANGLE :ANGLE + 1]
```



This spiral figure was made as a result of taking 720 steps. The first 360 steps created the upper right arm of the spiral, and the second 360 steps created the lower left arm. If we were to continue repeating these same commands, we would retrace the original figure.

This type of spiral is but one member of a large family of such curves. A procedure that lets us create this and other examples of closed spirals is as follows:

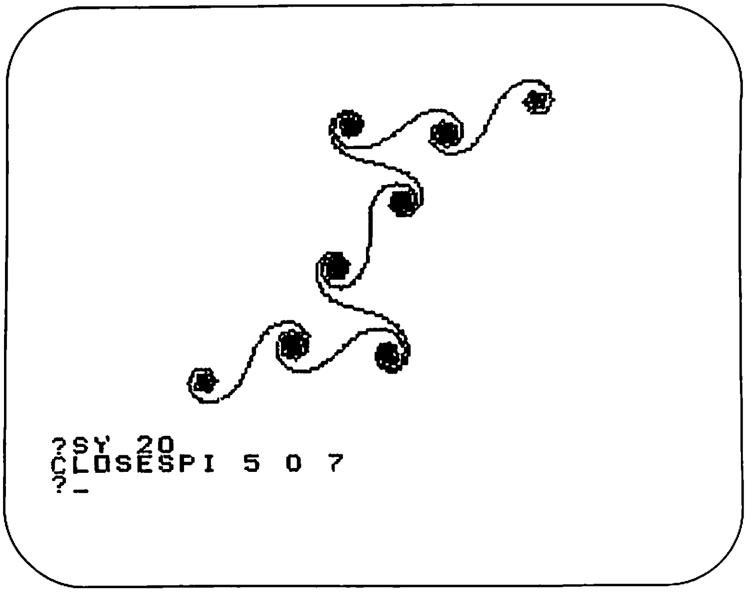
```
CLOSESPI :SIZE :ANGLE :INCREMENT
REPEAT 720 [FD :SIZE RT :ANGLE MAKE "ANGLE :ANGLE +
  :INCREMENT]
END
```

This procedure lets us create figures with different angle increments and with different starting angles. Our previous spiral can be drawn by entering

```
CS
SY 20
CLOSESPI 6 0 1
```

Instead of increasing the turning angle in 1-degree steps, suppose we choose another value—for example, 7 degrees. On first thought, we might expect to get a smaller version of the existing spiral, since we have changed only the increment by which the angle is changed. To see what happens, enter

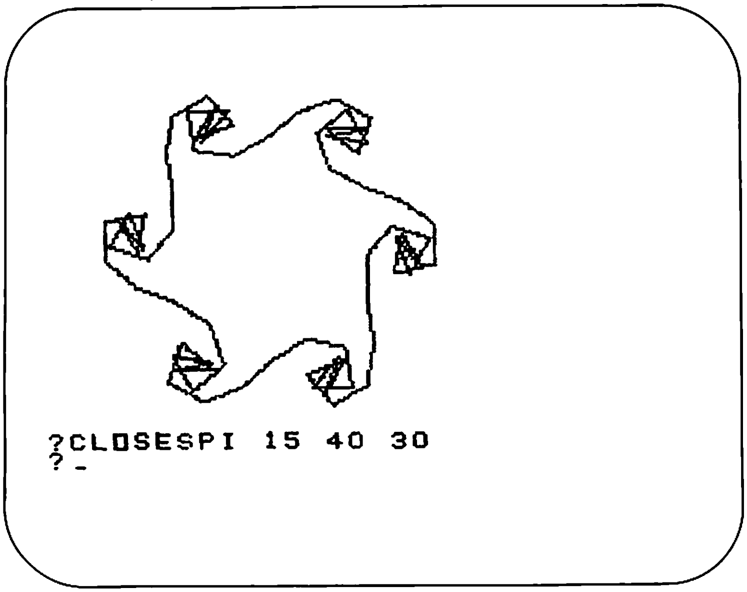
```
CS
SY 20
CLOSESPI 5 0 7
```

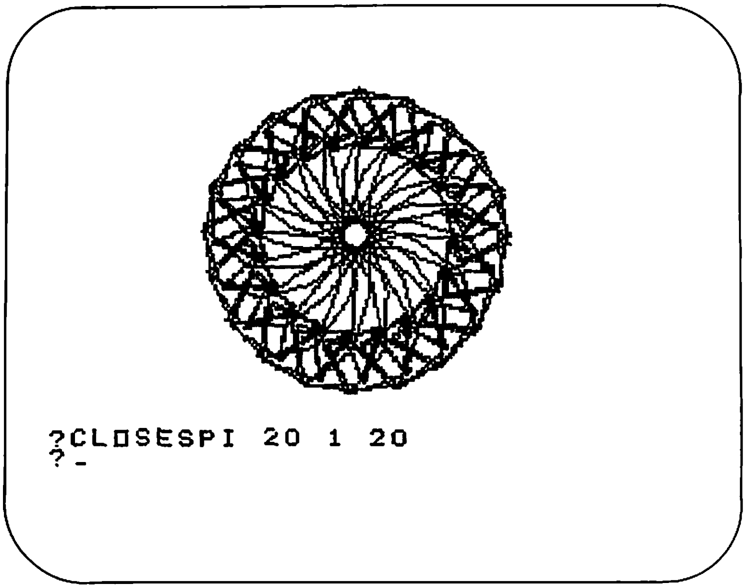


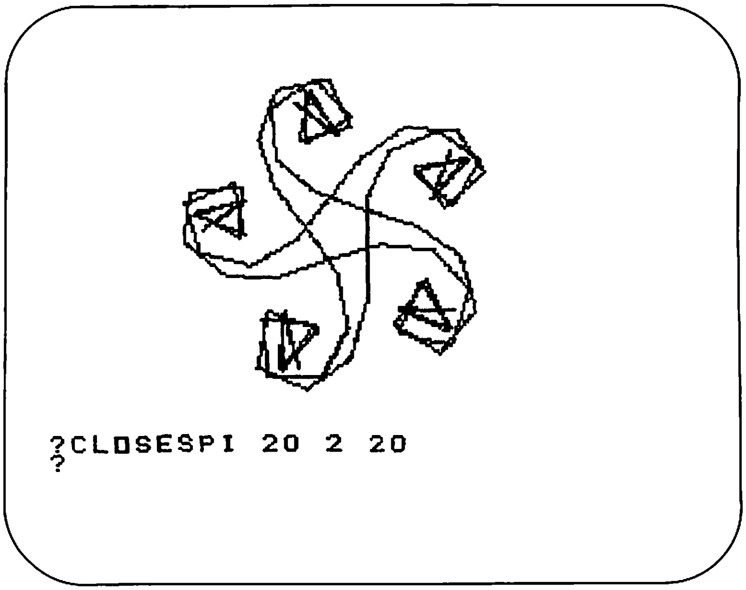
This surprising result occurs because the increment was chosen to keep us from reaching a turning angle of 180 degrees in the first arm of the spiral. As a result, the curve could not retrace itself until this condition was met.

More polygonal forms of the spiral can be made by starting with an offset angle. The following three figures were made using:

CLOSESPI 15 40 30
CLOSESPI 20 1 20
CLOSESPI 20 2 20




?CLOSESPI 20 1 20
? -


?CLOSESPI 20 2 20
?

You may wish to experiment with this procedure some more. It can create many beautiful pictures.

Open Spirals

Spirals can be created from arcs that are drawn so that each segment is larger than its predecessor by some ratio.

Spirals that grow by fixed ratios are commonplace in nature. They appear in snail shells, whirlpools, and numerous other natural objects and phenomena.

Spirals of this type can be approximated by a procedure that draws a series of 90-degree arcs, with each arc larger than its predecessor by a fixed factor. There are two types of spirals—those that turn to the left and those that turn to the right. To generate a right-handed 90-degree arc, we can use the RARC procedure:

```
TO RARC :SIZE
REPEAT 5 [FD :SIZE RT 18]
END
```

We can also use this procedure for generating right-handed spirals by creating the RSPIRAL procedure:

```
TO RSPIRAL :FACTOR :STEPS
MAKE :SIZE 1
REPEAT :STEPS [RARC :SIZE MAKE "SIZE :SIZE * :FACTOR]
END
```

To generate left-handed spirals, we can use the following procedures:

```
TO LARC :SIZE
REPEAT 5 [FD :SIZE LT 18]
END
```

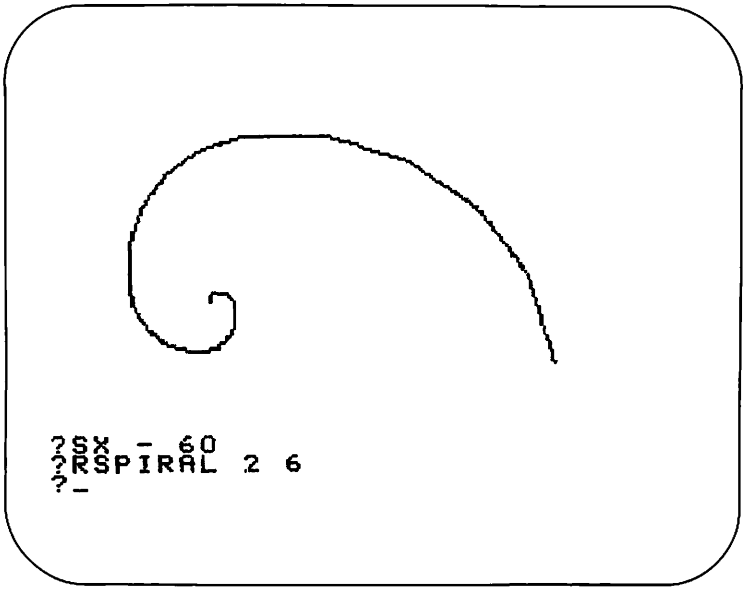
```
TO LSPIRAL :FACTOR :STEPS
MAKE :SIZE 1
REPEAT :STEPS [LARC :SIZE MAKE "SIZE :SIZE * :FACTOR]
END
```

Next, we can experiment with spirals that use different expansion factors. If you enter

```
CS
RSPIRAL 1 10
```

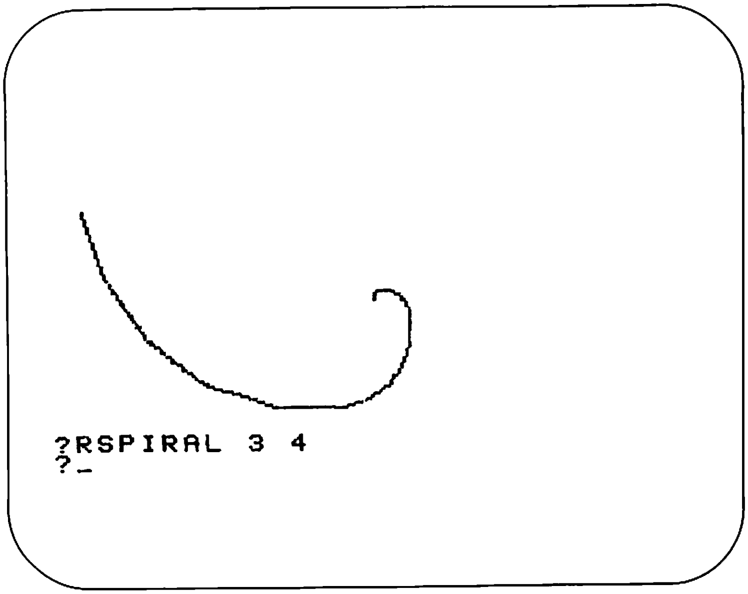
all you will see is a circle at the center of the screen. This is because you cannot increase the size of anything by multiplying it by 1. To see a spiral, we need a larger factor. Enter

```
CS
SX -60
RSPIRAL 2 6
```



By increasing the size of the expansion factor, we can create spirals that are more open. As an example, enter

```
CS
RSPIRAL 3 4
```

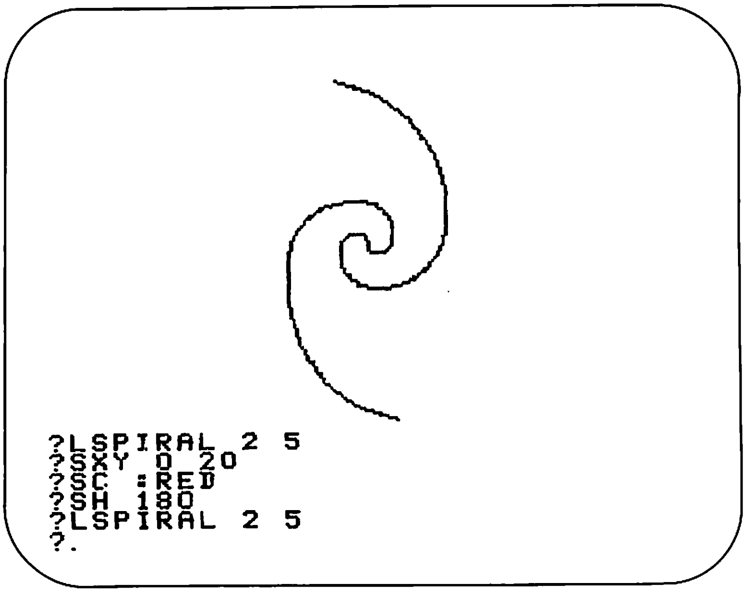


Interlaced spirals can be made by turning the turtle by different amounts before starting the spiral. As an example, enter

```
CS
SY 20
LSPIRAL 2 5
```

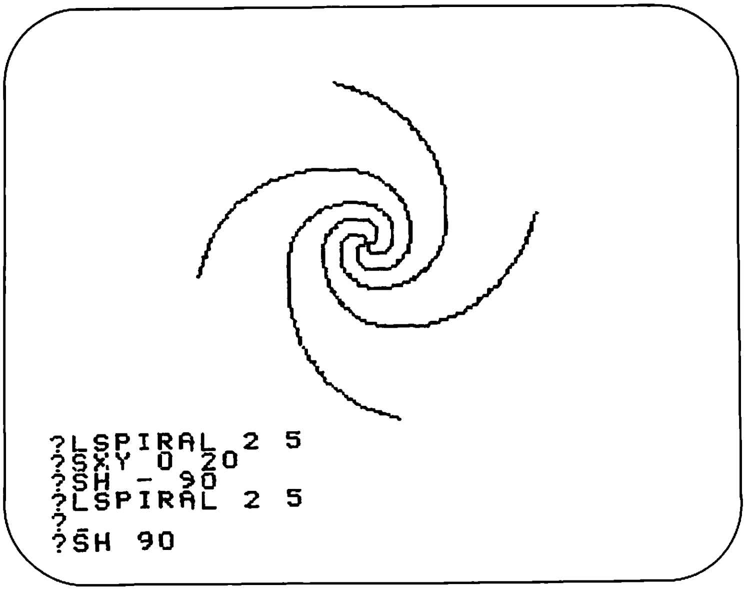
This draws a left-handed spiral from the center of the screen. Next, we will change the pen color to red and draw the same spiral from the center, but we will turn the turtle by 180 degrees first. Enter

```
SXY 0 20
SC :RED
SH 180
LSPIRAL 2 5
```



Note that the blue spiral lies near the middle of the gap in the white spiral. To produce interwoven spirals with different locations, the same process can be repeated with other starting angles. The following example will add an orange line on either side of the red one:

```
SC :ORANGE
SXY 0 20
SH 90
LSPIRAL 2 5
SXY 0 20
SH -90
LSPIRAL 2 5
```



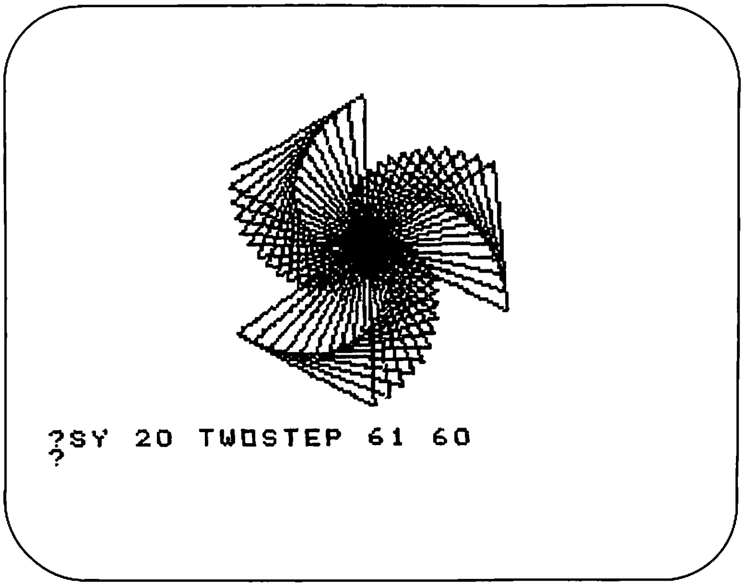
Projects that Involve Growth

Now that we have explored three types of spiral figures, you should explore some on your own. As an example, try changing SQUIRAL to increase both the size and the angle at the same time. See how pleasing the resulting figures are to you.

Next, you might wish to explore the properties of a procedure that moves the turtle back one step for each two steps forward (turning the turtle by some angle in between). One procedure for doing this is as follows:

```
TO TWOSTEP :ANGLE :STEPS
MAKE "SIDE 0
REPEAT :STEPS [FD :SIDE RT :ANGLE BK :SIDE /
  2 RT :ANGLE MAKE "SIDE :SIDE + 2]
END
```

The following figure was generated with SY 20 TWOSTEP 61 60.



Try some other projects. As an example, modify SQUIRAL (with judicious use of PENUP and PENDOWN) to have it only draw dots at the ends of each line. Replace the dots by small stars and see how you like the result.

Spiral procedures can generate many interesting pictures!

VIII. Additional Drawing Aids

As mentioned in the preceding chapter, you now have all the tools you need to create many wonderful graphic compositions. You may find, however, that it is tedious to use SXY and SH all the time to pick the starting points for your procedures. Wouldn't it be nice if we could just move the turtle to some starting position and orientation without having to bother with a lot of typing? For that matter, wouldn't it be nice if some of our procedures could be drawn in response to a single keystroke?

This chapter introduces two new ways for you to communicate with your computer—using a joystick and having Logo "read" the keyboard in the middle of executing a procedure.

Before starting, clear your workspace of all procedures except such basic building blocks as POLY, STAR, and CIRCLE.

A Thing of Beauty Is a Joystick Forever

You may have noticed that the left side of your TI-99/4A computer console is equipped with a multipin jack. This connector is designed to accept a pair of joystick controllers made just for this computer. To use the joysticks, you need to do two things: you must plug them in to the computer, and you must learn which Logo words will let the computer monitor the joysticks' positions.

Connecting the joysticks requires simply plugging them in. Since the connector will only fit one way, proper installation of the joysticks, should be completed in seconds. The status of the joysticks can be determined with the Logo word JOY. To see how this word works, enter the following procedure:

```
TO JOYTEST
PRINT SENTENCE JOY 1 JOY 2
JOYTEST
END
```

JOY 1 is a Logo function that returns a number associated with the position of joystick 1, and JOY 2 is the corresponding function for the second joystick. This procedure prints a message on the display screen consisting of a "sentence" (actually, a list) containing the words that provide the position values for the two joysticks. Note that PRINT prints only one word or list at a time. If you want to print two words side by side, you must combine them into a single list first.

JOYTEST has another interesting feature. After printing out the positions of the two joysticks, it uses itself again.

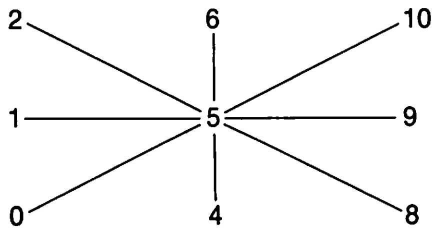
The task of having a procedure use itself is called *recursion*. TI Logo keeps track of each time a procedure is used but not completed. Since we never get to the end of JOYTEST, we will run out of computer memory sooner or later. We will soon learn a way to avoid this problem, but for our present needs, JOYTEST works perfectly. Enter

```
NOTURTLE CS
JOYTEST
```

You will see two columns of numbers printed on the screen. The first column is the number associated with the position of joystick 1, and the second is the number associated with the position of joystick 2. If you are not pushing or pulling one of the joystick handles, both numbers should be 5.

So far, we don't know which joystick is number 1 and which is number 2. Pick up either one and hold it in one hand, with the cable going away from you. If you grab the handle with the other hand and push or pull it, you will see the numbers on one of the columns change. If the numbers change in the first column, you are holding joystick 1; otherwise, you are holding joystick 2. You should label your joysticks with tape so that you can tell them apart.

As you move the joystick handle around, you will see a pattern to the numbers that appear, as follows:



Both joysticks produce the same pattern of numbers. Since each position of the joystick is associated with a unique number, we can use this number to move the turtle around on the screen.

As an example, let's create a procedure for joystick 1 that moves the turtle forward if it is moved up and turns the turtle to the left or right as the joystick is pushed in either of these directions. We will have the procedure stop if the joystick handle is pulled toward you. The following procedure will do all these things:

```
TO MOVE
PU
START:
MAKE "STK JOY 1
IF :STK = 6 THEN FD 5
IF :STK = 1 THEN LT 15
IF :STK = 9 THEN RT 15
IF :STK = 4 THEN PD STOP
GO "START
END
```

Because we have introduced several new Logo words in this procedure we will explain them before we try out the procedure. First, note the word START:. This word is a Logo *label*. Labels are used in conjunction with the GO command to cause instructions within a procedure to be used over and over again without having the computer run out of memory. When Logo executes the command GO "START, it causes the procedure to start execution over again at the line containing the label. You can choose your own label names, of course, but you must remember to end the label with a colon, or Logo will treat it like a procedure name. Also note that the colon does *not* appear in the GO command; a quotation mark is used instead.

The second new command is IF (relation) THEN (instruction list). When Logo encounters an IF command, it evaluates the expression that follows the word IF. If the relation is true, Logo executes the rest of the line. If it is false, Logo jumps to the next line of instructions.

The last new word we used was STOP. When Logo encounters STOP, it halts the execution of the procedure and returns control to the user or to the procedure that used the procedure that just stopped. If MOVE, for example, was used by another procedure (such as DRAW), then when MOVE stops, the procedure DRAW picks up where it left off.

Let's try MOVE to see how it works. Enter

```
TELL TURTLE CS
MOVE
```

As you push the joystick forward, the turtle starts to move up the screen. If you push the joystick to the right or left, the turtle will turn in the corresponding direction in 15-degree increments. Finally, pull the joystick back toward you. This will stop the procedure.

Single-Keystroke Procedures

Now that we know how to move the turtle around on the screen with the joystick, let's examine a way to create various figures by pressing just a single key.

Logo provides an easy way to read characters from the keyboard using the word READCHARACTER (or RC for short). To see how this command works, enter the following procedure:

```
TO TYPEIT
MAKE "CHAR RC
TYPE :CHAR
TYPEIT
END
```

If you now enter

```
NOTURTLE
TYPEIT
```

and start typing, you will see text appear on the screen. When TYPEIT encounters the RC command, it waits for you to press a key. As soon as you press a key, the character you pressed is placed in the variable CHAR. Next, this character is displayed on the screen with the TYPE command. The only difference between TYPE and PRINT is that PRINT starts on a new line each time it is used but TYPE does not.

To run procedures with single keystrokes, we first have to define our procedures to have single-character names. As an example, let's define the following procedures:

```
TO S
POLY 40 4
END

TO P
STAR 40 5 2
END

TO C
CIRCLE 5 10
END
```

Now we can combine all we have learned into a simple drawing program that moves the turtle under joystick control and lets us draw figures with single keystrokes. The procedure DRAW is one simple way to do this:

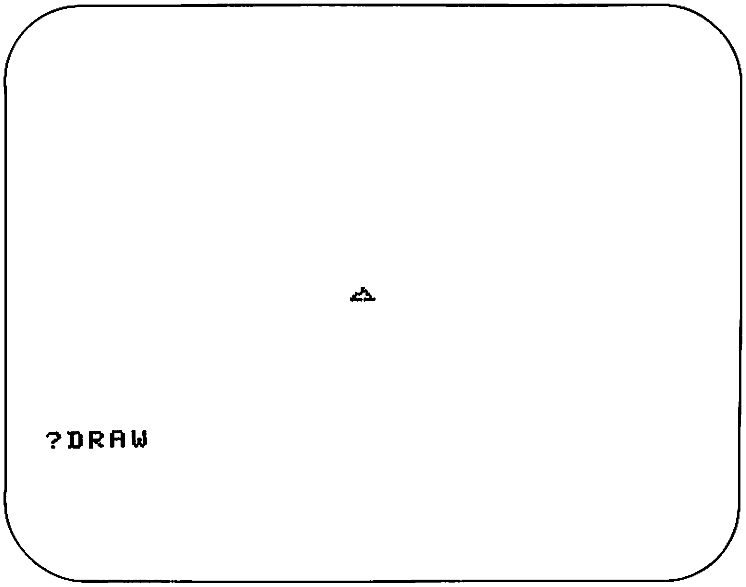
```
TO DRAW
START:
MOVE
MAKE "CHAR SE RC [ ]
REPEAT 1 :CHAR
GO "START
END
```

Because REPEAT needs a list as input, we can't just use the character returned from the RC command without making it into a list first. We do this with the SENTENCE (or SE) command. Since SE makes a list from two words or lists, we use the character given by RC as the first word and the empty list, [ ], as the second. If you find this is getting too technical for your blood, don't feel that you must understand it perfectly—it works even if you don't fully understand it.

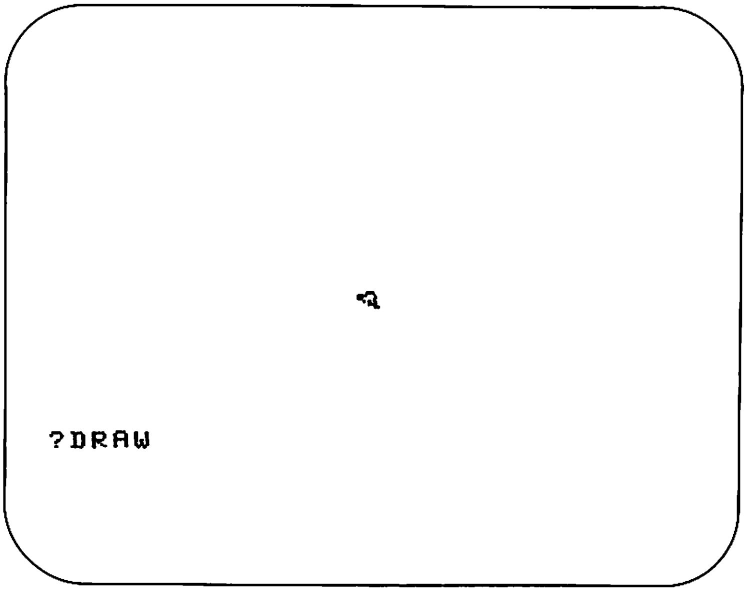
Without further ado, then, enter

```
TELL TURTLE CS
DRAW
```

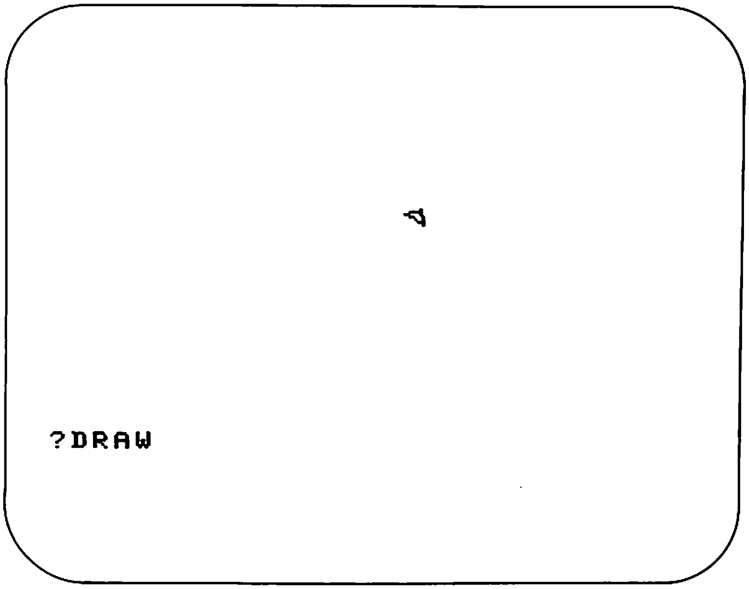
We will start with the turtle in the center of the screen.



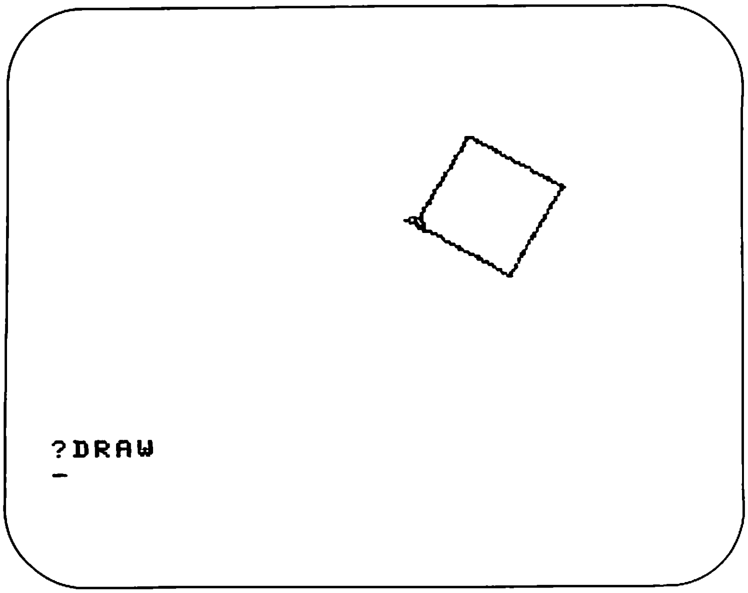
Now turn the turtle a bit to the right with the joystick.



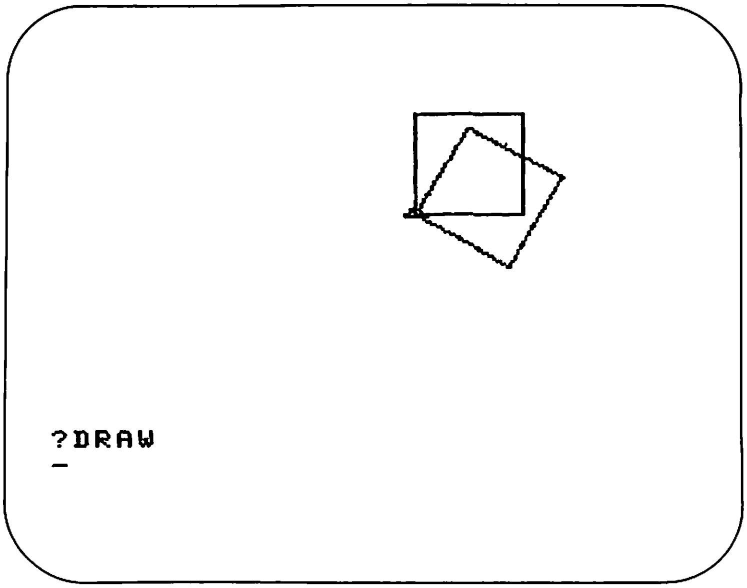
Push the joystick forward to a new location.



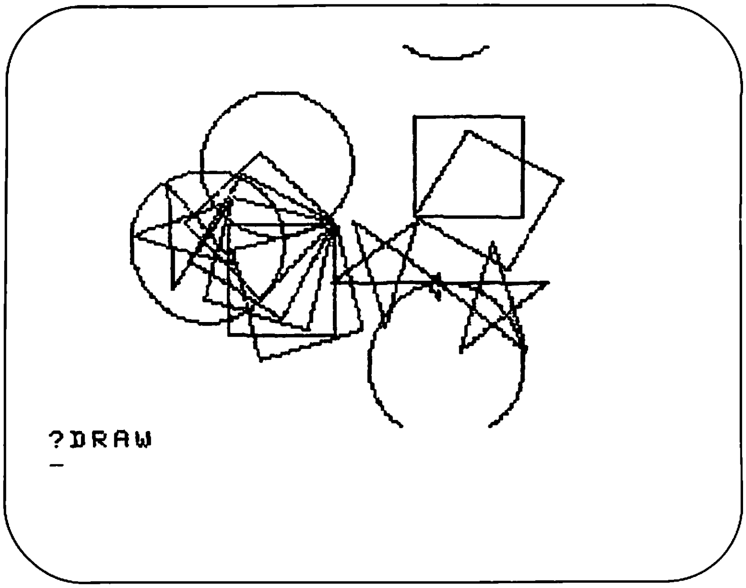
Next, pull the joystick back and press the S key. A tilted square will be drawn on the screen.



If you now push the joystick to the left, the turtle will turn to the left. When the turtle is pointing straight up again, pull the handle and press S again.



Using the joystick to steer and move the turtle, add some more figures to this picture.



As you can see, we have used Logo to create a simple graphics *program* that lets us "rubber-stamp" procedures at different screen locations. Using the ideas we have explored in this chapter, add some other features to this program to make it useful for your purposes. Some possibilities would be to create single-keystroke procedures to change pen color or background color and to draw different shapes.

Some Projects

At this point in our exploration, we not only have covered turtle graphics, but we have shown how to create your own simple drawing programs. Because we don't have an easy way of saving the artwork from DRAW on the disk for future use, it may be more valuable to think of this program as a sketching tool that lets you try out graphic ideas before incorporating them in a complete work.

The next few chapters focus on the other powerful graphic capabilities of TI Logo, including animation. This is a good time for you to examine your feelings about computer graphics once again. Do you like creating pictures with turtle graphics? Do you find that you are expressing different ideas with this medium than you would with watercolors, for example?

You might want to spend a week or so exploring some picture ideas on your own before proceeding to the next chapter. Can you make a harsh picture using only straight lines? Can you make a warm picture with arcs and circles? Next, try reversing things: make a harsh picture using only arcs and circles and a warm picture using only straight lines. Don't forget that you can use color in your pictures; color is a powerful tool in itself.

IX. Tiles—Another Logo Graphics Tool

We have spent so much time exploring Logo's turtle graphics that we have totally ignored other ways that TI Logo lets us create pictures. One of these ways uses the TI character set.

At first glance, you might think that characters are too limiting for graphic compositions. After all, there are only 26 letters and a similar number of numerals and punctuation marks. These hardly seem to be likely candidates for graphic composition. In addition to the apparent limitation on the number and type of characters, the TI-99/4A display screen can accommodate only 24 rows of characters, with no more than 32 characters in a line. This means that we can place a given letter, such as A, in any of 32 columns or 24 rows.

Suppose, however, that we could change the letter A so that it had some other shape—that of a bird, for example. Then, even with a limited number of display positions, we could create some interesting patterns on the screen.

TI Logo has a powerful set of commands that let us change the shape of any displayed character and let us define the shapes of more than 100 characters that are otherwise used by TI Logo only when it is creating pictures with turtle graphics. Once you have defined new shapes for your characters (they are called *tiles*), you can save them on your disk along with any Logo procedures that use them.

The rest of this chapter is devoted to exploring the patterns we can make with tiles.

Placing Tiles on the Screen

Each tile in TI Logo is assigned to a number between 0 and 255. If we want to place a particular tile on the screen, we need to know the number associated with it. The following table shows the numbers associated with the characters we normally see on the screen:

GROUP 1		GROUP 2		GROUP 3	
character	value	character	value	character	value
(space)	32	(	40	0	48
!	33	)	41	1	49
"	34	*	42	2	50
#	35	+	43	3	51
$	36	,	44	4	52
%	37	−	45	5	53
&	38	.	46	6	54
'	39	/	47	7	55

GROUP 4		GROUP 5		GROUP 6	
character	value	character	value	character	value
8	56	@	64	H	72
9	57	A	65	I	73
:	58	B	66	J	74
;	59	C	67	K	75
<	60	D	68	L	76
=	61	E	69	M	77
>	62	F	70	N	78
?	63	G	71	O	79

GROUP 7		GROUP 8	
character	value	character	value
P	80	X	88
Q	81	Y	89
R	82	Z	90
S	83	[	91
T	84	\	92
U	85	]	93
V	86	^	94
W	87	_	95

Each of these characters has been placed in one of eight clusters. The reason for this clustering will be given later.

What is meant by the character values shown in the table? Suppose you want to place an ampersand (&) on the screen. One way to do this is to enter

```
TYPE "&
```

This types an ampersand but does not move the cursor to the next line. You can also print an ampersand by using the command

```
PRINTCHAR 38
```

If you check in the table, you will see that the symbol & has the value 38.

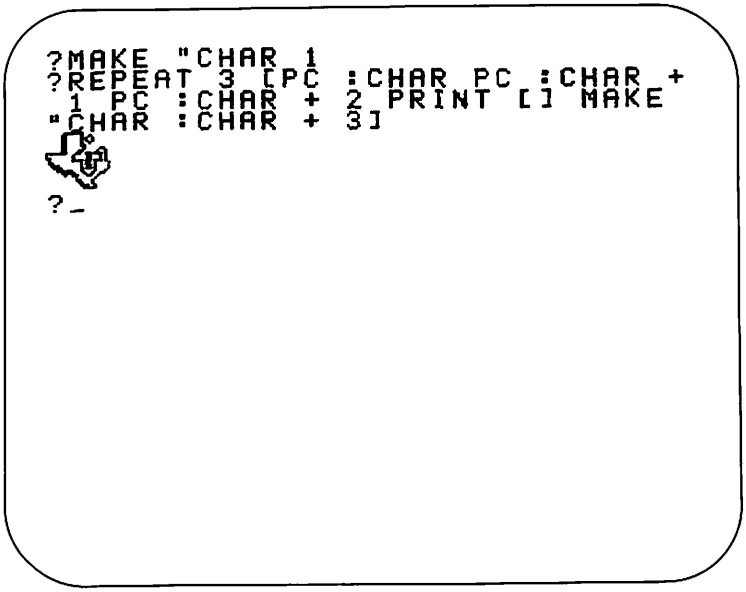
Since the PRINTCHAR command produces the same result as the TYPE command, you might ask why PRINTCHAR (or its short form, PC) is necessary. The power of PRINTCHAR is that it can be used with any of the 255 available tiles—not just with those associated with the alphabet, numerals, and punctuation marks.

To see some of the tiles that have already been defined, enter

```
NOTURTLE CS
MAKE "CHAR 0
REPEAT 128 [PC :CHAR MAKE "CHAR :CHAR + 1]
```

The screen will display the numerals, letters, and punctuation marks, preceded by several spaces and a few characters that look like random garbage. These strange-looking characters are really a set of tiles used for the TI emblem in the display you see when the computer is first turned on. To see this emblem again, enter

```
CS
MAKE "CHAR 1
REPEAT 3 [PC :CHAR PC :CHAR + 1 PC :CHAR + 2
  PRINT [  ] MAKE "CHAR :CHAR + 3]
```



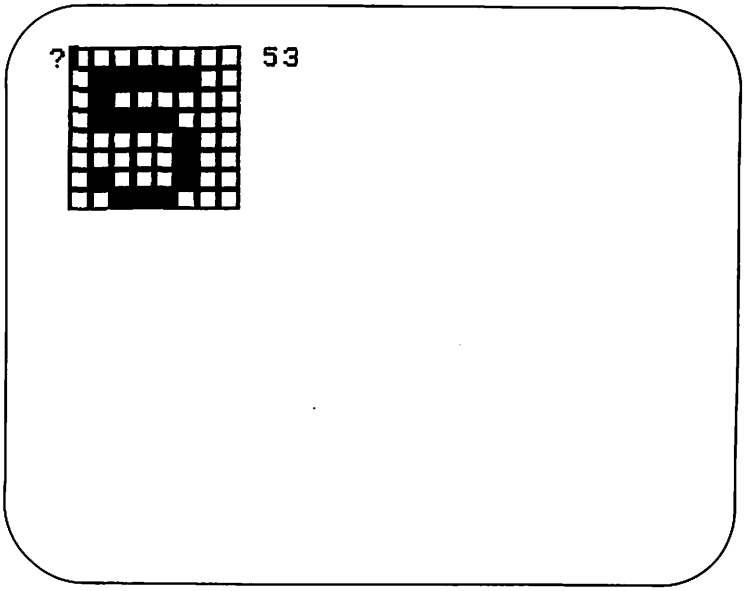
This set of commands prints a little map of Texas with TI in it. Notice that we used PRINT [ ] to print an empty list and force the cursor to the next row.

This TI symbol is a simple example of a picture made with tiles. Before showing another way of placing tiles on the screen and showing how to change tile colors, we will do something even more powerful—we will show how to design your own tile patterns.

Making Your Own Characters

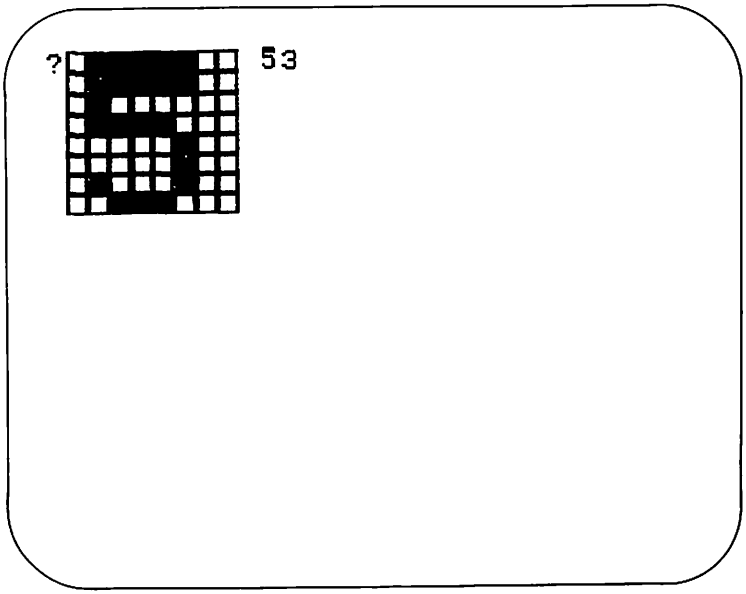
One of the most powerful features of TI Logo's character tiles is that you can make them take any shape you want. To change the shape of a character, you first must type MAKECHAR, followed by the number of the character you want to change or define. To show how flexible this editing system is, we will demonstrate it by editing the shape of a numeral and then allowing the computer to give us a strange result to a calculation. Enter

```
CS
MAKECHAR 53
```



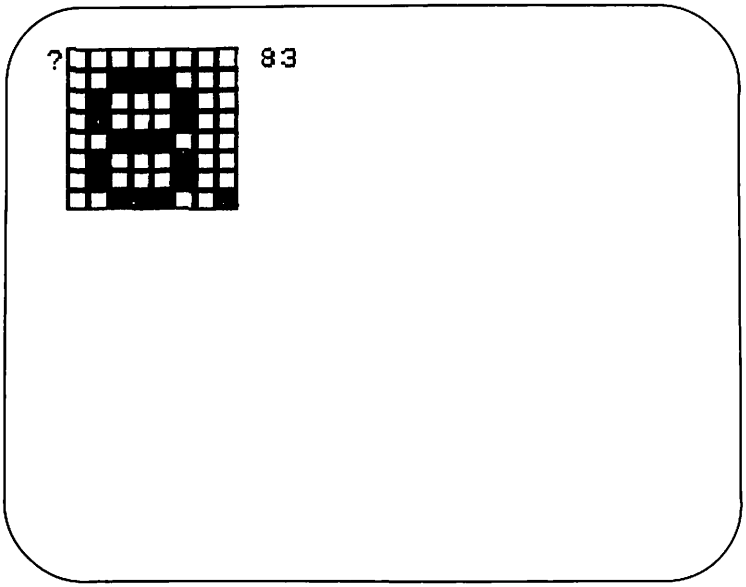
As soon as you press ENTER, the screen will turn green, and a large number 5 will appear in the upper left corner, obscuring the word MAKECHAR. The number 5 is inside a grid of 8 × 8 small squares. This 8 × 8 grid is common to all tiles. You can change each small square to be filled or unfilled, as you wish, and thus change the shape of the character in the tile.

To change a character shape, you use the same four arrow keys that are used in editing procedures. There is an important difference in the way these keys are used, however. If you press the arrow key by itself, it will leave an empty square. If you hold down the FCTN key while pressing the arrow key, it will leave a filled square. To experiment with this, press the right arrow key once. The cursor moves to the right by one space and leaves the first space blank. Next, hold down the FCTN key and press the right arrow key five times.



You will see the top bar of the number 5 get thicker, both in the large 8 × 8 grid and on the 5 in the numeral 53 shown on the screen. As you change the shape of a character, these changes appear in the 8 × 8 matrix as well as in any instances of this character that may be displayed on the screen at the same time.

Next, use the arrow keys to change the tile from a 5 to an 8. When you are done, the image should look like this:



You can now go back to the text screen by pressing BACK. The first thing you might notice is that the number 53 on the screen reads 83! If you press the number 5, you will see an 8. To see how silly we can get, enter

```
PRINT 2 + 3
```

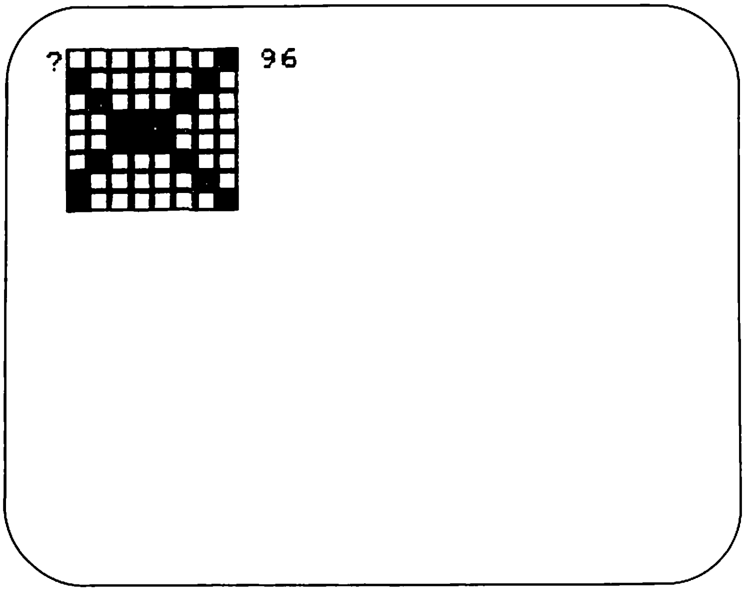
When you press the return key, you will see an 8 on the screen. The reason for this error (most of us expect 2+3 to equal 5) is that we have defined the shape of the numeral 5 to be the same as that of the numeral 8. As far as the computer is concerned, the shape of a character has nothing to do with its meaning.

Just to avoid confusion, press QUIT and start Logo all over again. This will reset the tiles to their original shapes.

Now that we know how to edit tiles, let's experiment with the creation of some wallpaper patterns. As an example, enter

```
CS
MAKECHAR 96
```

You should see a blank 8 × 8 grid in which to define a new tile. Using the arrow keys (and the FCTN key), define a shape that looks like this:

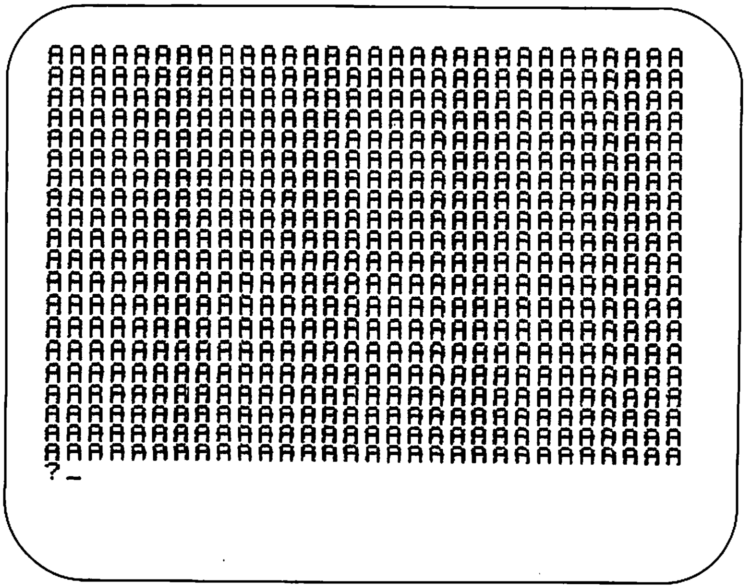


Next, press BACK and define this new procedure:

```
TO TILES :NUMBER
CS
REPEAT 600 [PC :NUMBER]
END
```

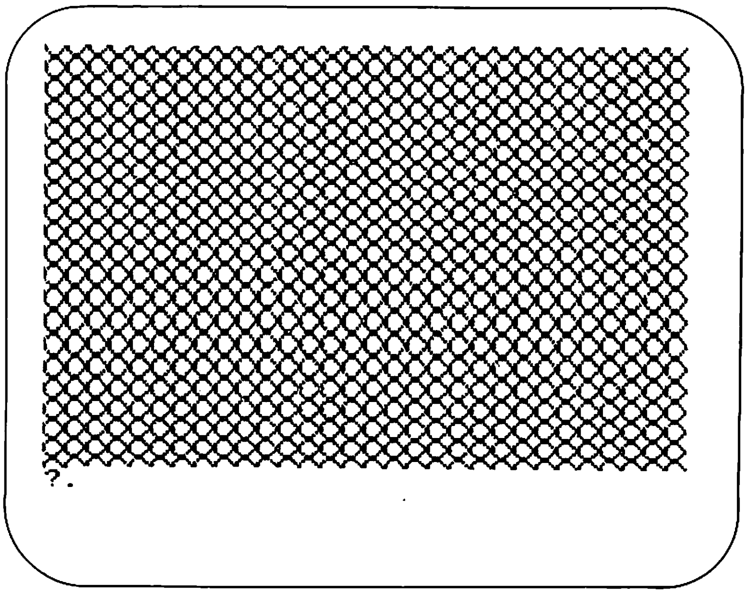
This procedure will fill most of the screen with images of any tile you select. To see a screen full of A's, for example, enter

```
TILES 65
```



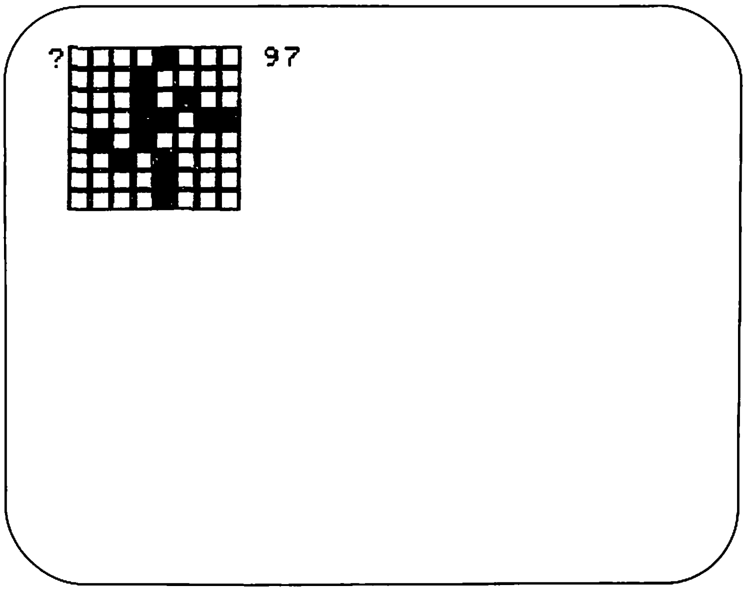
To see our new pattern, enter

TILES 96



This pattern is an interlocking network made by repeating one character.

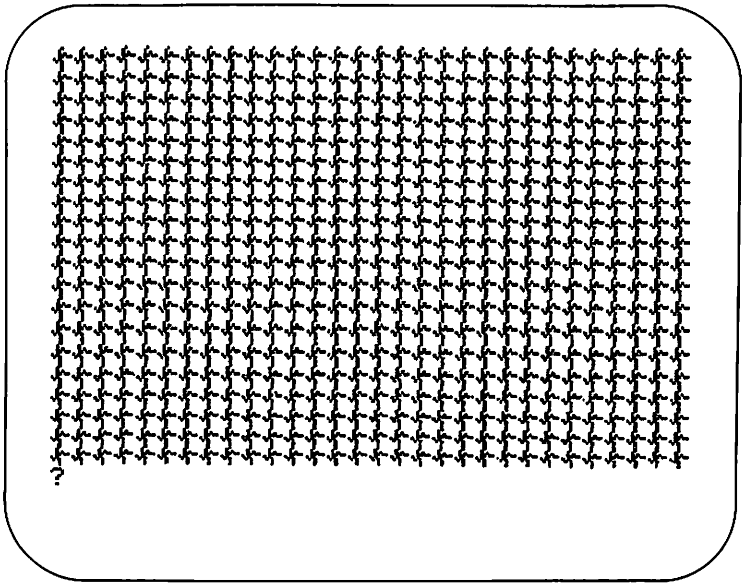
Experiment with different tile shapes to see what patterns you can create. As an example, if you define character 97 to have this shape,



then when you type

TILES 97

you will get this pattern:



(*Note:* When you define your own tile shapes with character values greater than 95, you must *not* enter the turtle graphics mode without saving your tiles on the disk first. Each time you enter TELL TURTLE, the computer erases any shapes you have defined for characters from 96 to 255.)

Coloring and Placing Tiles

Getting a tile to change colors is a fairly straightforward process. To see how it is done, first enter

```
CS
PRINT "ABCDEFGHIJKLMNOP
```

This prints a row of character tiles on the screen. Next, suppose we wanted to change the color of the letter D (character 68). To do this, enter

```
TELL TILE 68
SETCOLOR :RED
```

As soon as you press ENTER, you will see all the letters from A through G change color to red! The reason this happens is that TI Logo, for purposes of assigning colors, groups characters into clusters of eight. When you change the color of one character in the cluster, you change all the others as well. This is why the characters in the earlier table were grouped in clusters of eight.

There is only one more command you need in order to master tile graphics. This command lets you place any character in any screen position you want. Our previous command, PRINTCHAR, is very useful when you want to pack characters next to each other in a sequence, but it isn't very useful if you want to place a character at some absolute screen position. Also, if you counted carefully, you may have noticed that PRINTCHAR puts characters in only 30 of the 32 available columns.

The command that lets us put characters anywhere on the screen is PUTTILE (or PT). The format for the PUTTILE command is PUTTILE (character column row). The character number is the same one we have been using. The column numbers run from 0 to 31 as you move from left to right on the screen, and from 0 to 23 as you move from top to bottom. (Depending on how your display is adjusted, you may not see characters in columns 0 or 31.)

To see how PUTTILE works, let's place a tile in the 16th column and row. Enter

```
CS
PT 97 16 16
```

This will cause a character to appear near the center of the bottom half of the screen. If you now enter

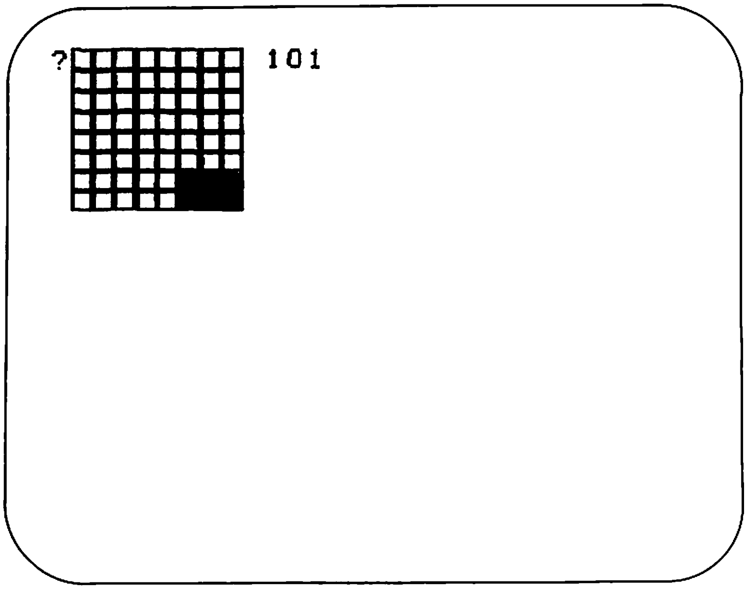
```
PT 96 16 16
```

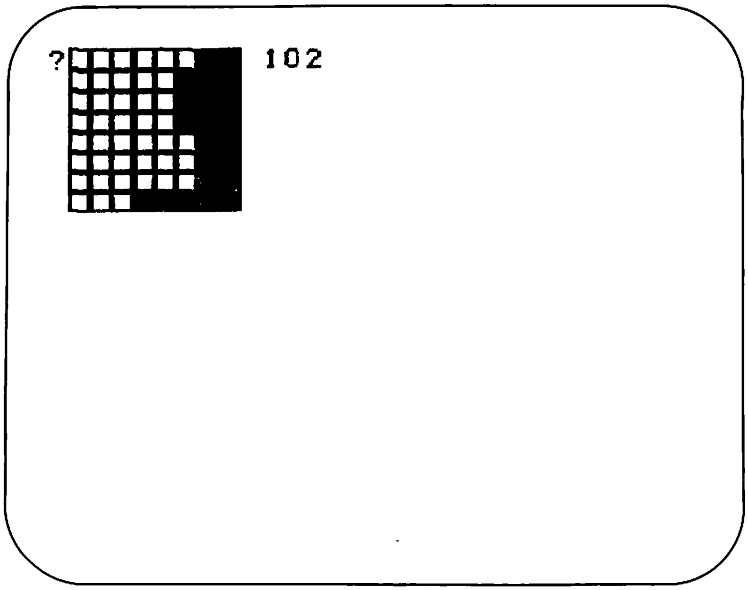
this character will be replaced by the character whose value is 96.

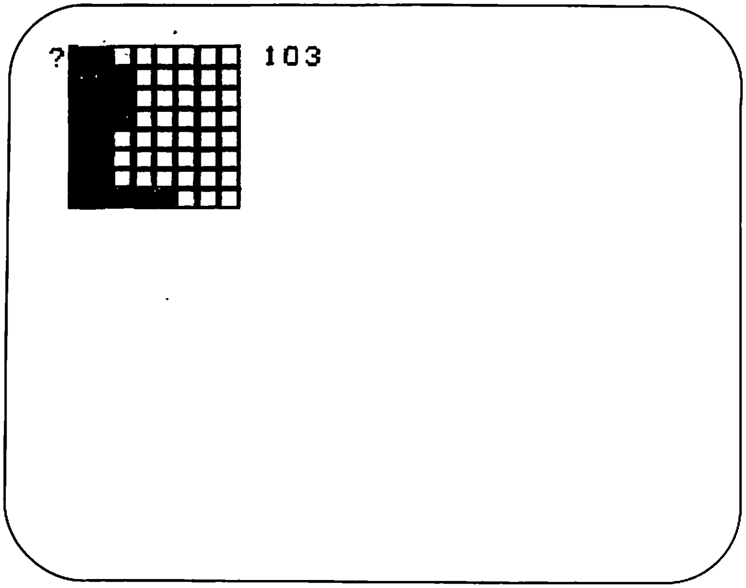
The limitation of an 8 × 8 matrix can be quite restricting, but by using PUTTILE, we can create larger matrices from 8 × 8 building blocks. As an example, you might want to create a design on a 32 × 32 grid. To make your design, mark off a 32 × 32 grid on a sheet of paper (quadrille ruled paper is perfect for this application). Next, shade in your pattern on the grid. Remember that each square in the grid is either completely filled or completely empty. Next, divide the 32 × 32 grid into sixteen 8 × 8 square arrays. Each of these arrays represents one tile. Use MAKECHAR to define each of these 8 × 8 subunits as a different character (for example, characters 101 through 116). You can then display your finished picture by using a procedure such as this one:

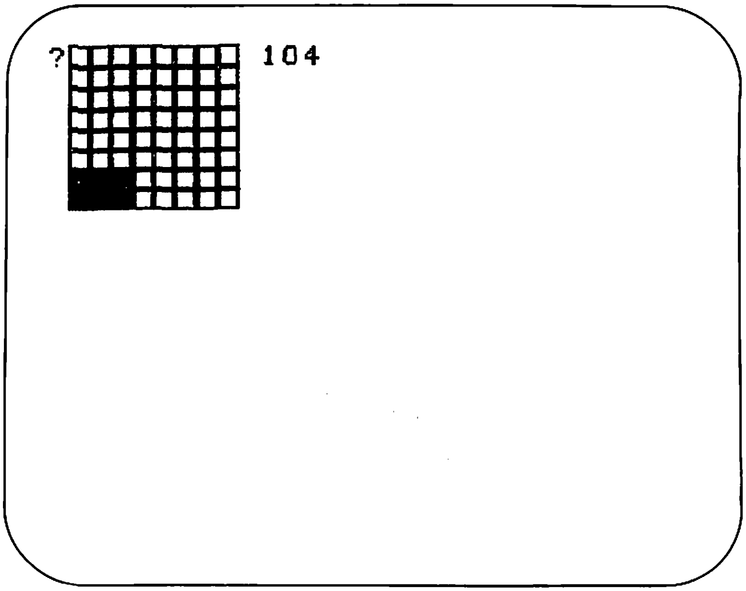
```
TO DISPLAY
PT 101 12 12 PT 102 13 12 PT 103 14 12 PT 104 15 12
PT 105 12 13 PT 106 13 13 PT 107 14 13 PT 108 15 13
PT 109 12 14 PT 110 13 14 PT 111 14 14 PT 112 15 14
PT 113 12 15 PT 114 13 15 PT 115 14 15 PT 116 15 15
END
```

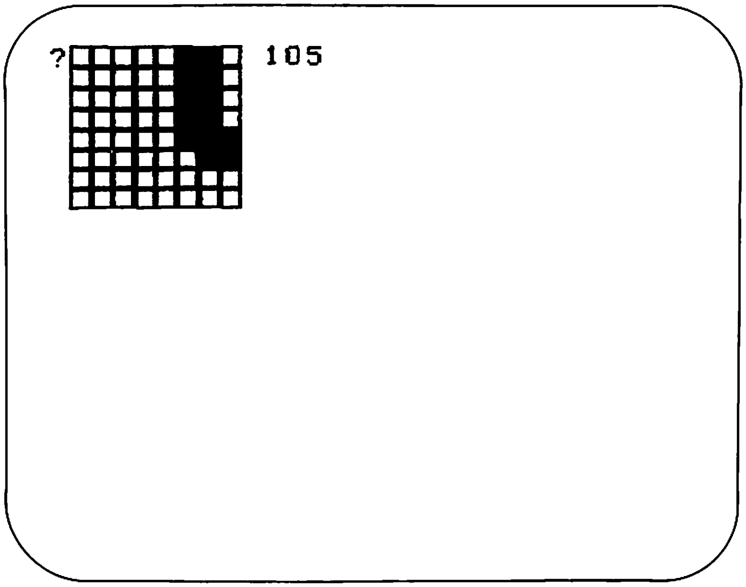
As an example of what can be accomplished with this graphic technique, place the following patterns in tiles 101 through 116, in order:

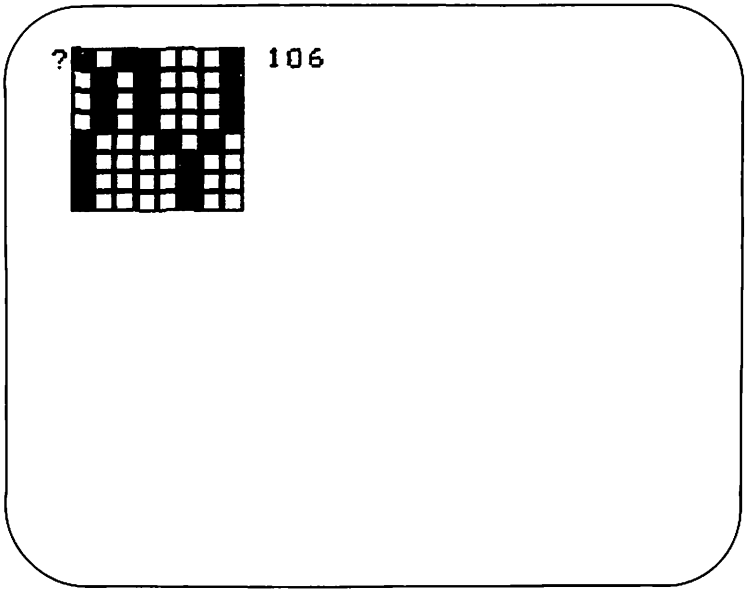

?
101

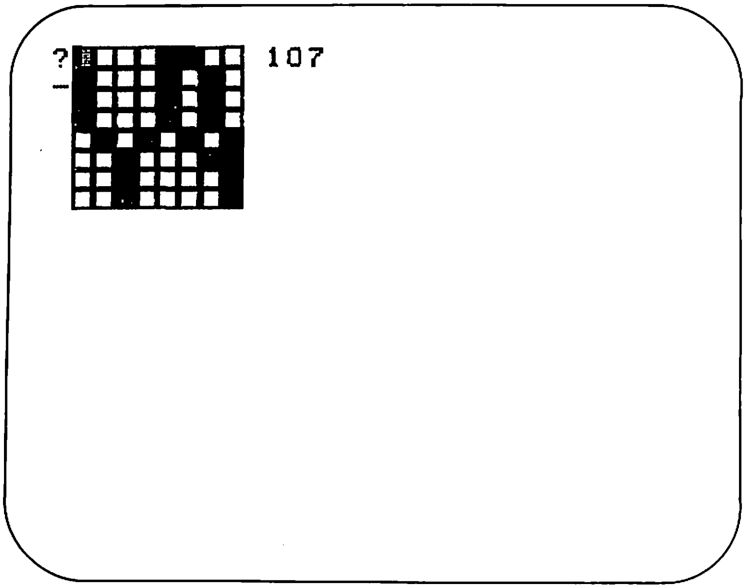

?
102

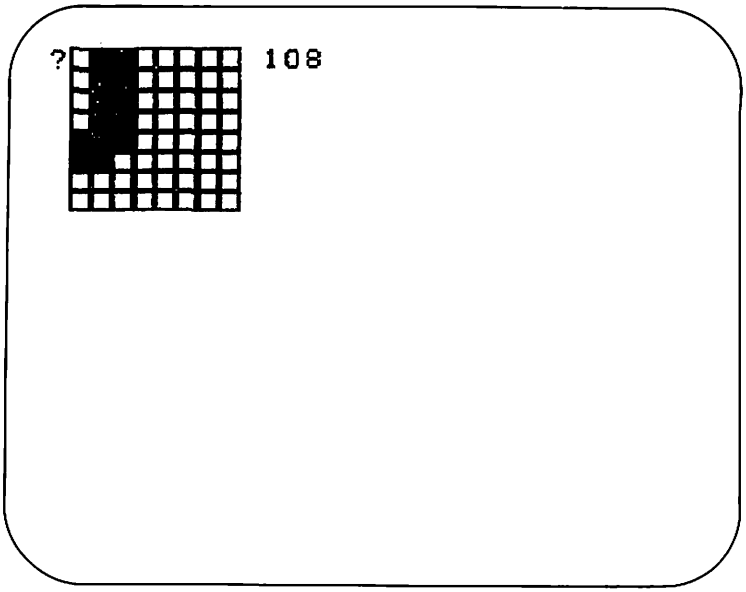

?
103

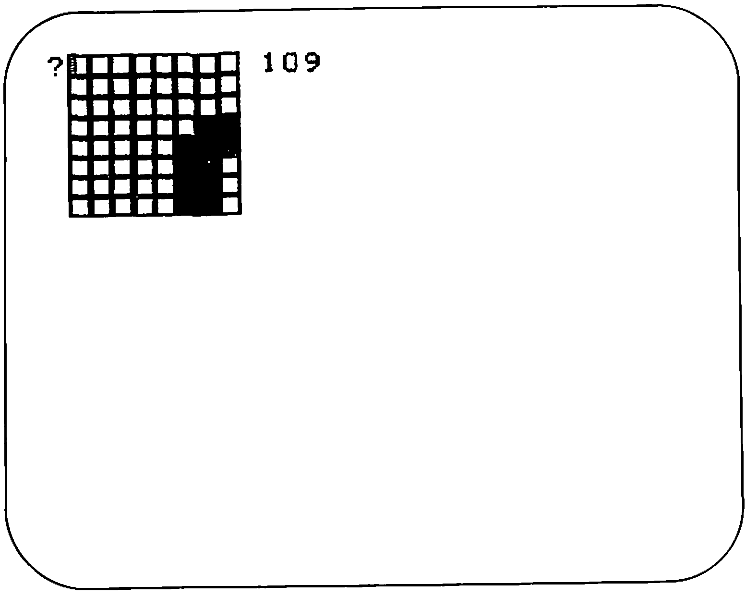

?
104

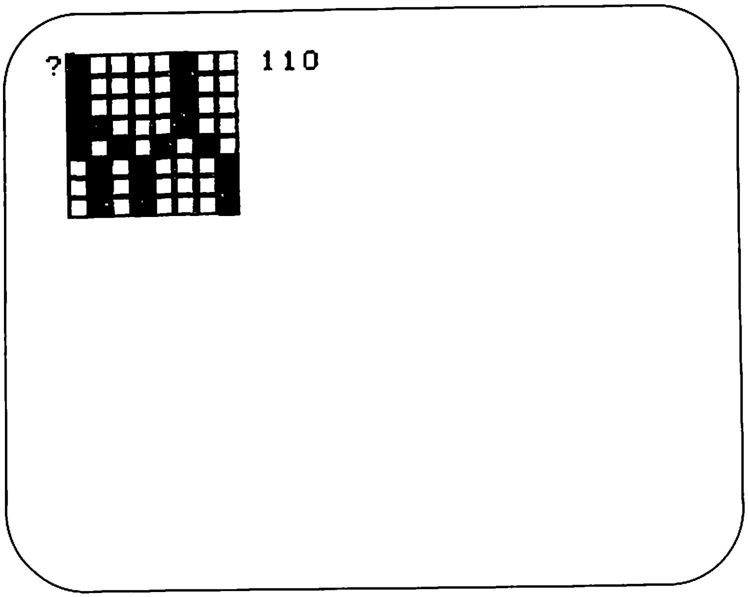

?
105

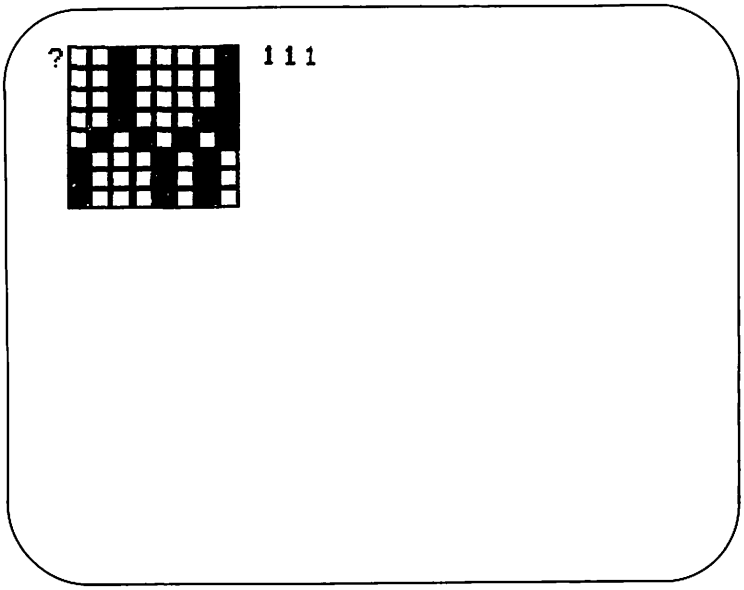

?
106

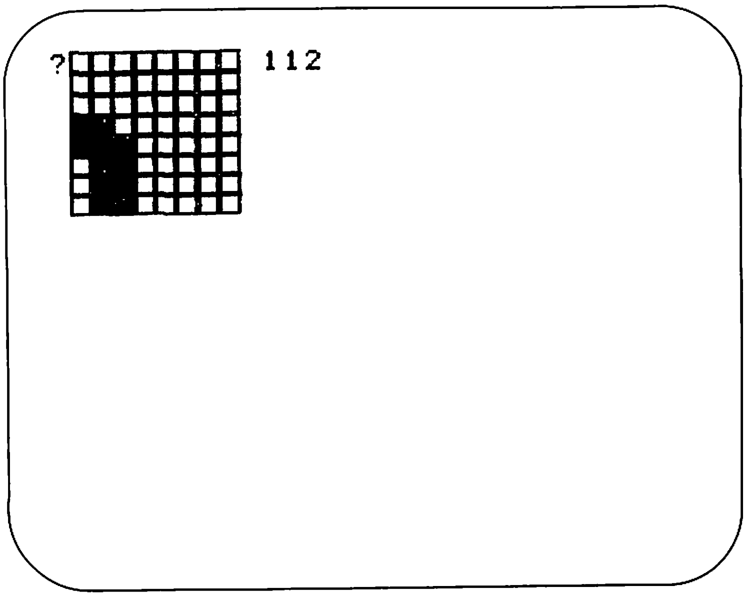

?
107

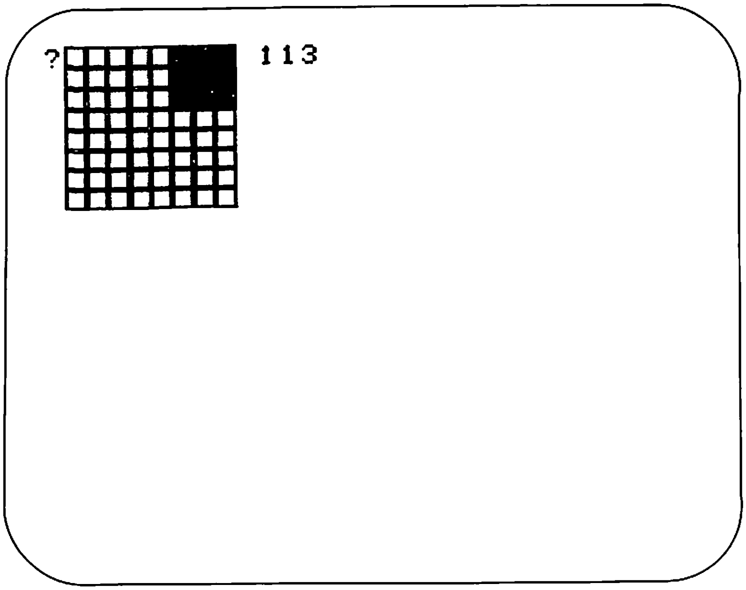

?
108

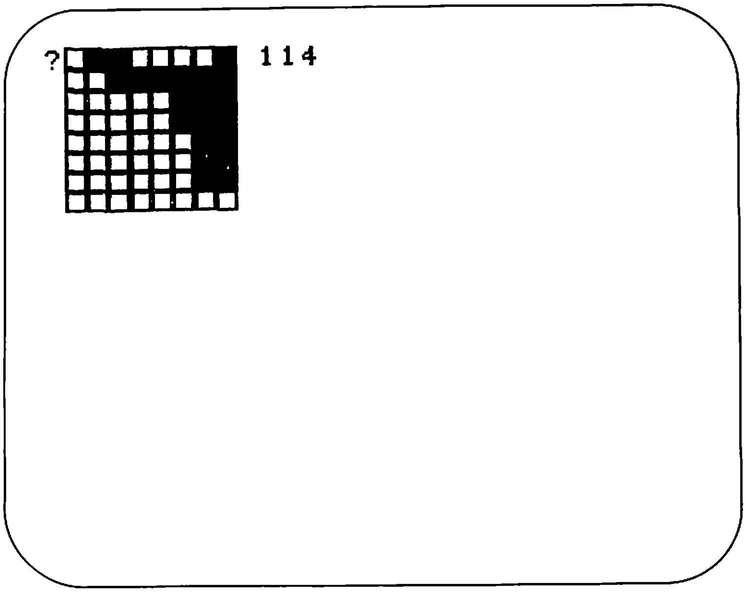

?
109

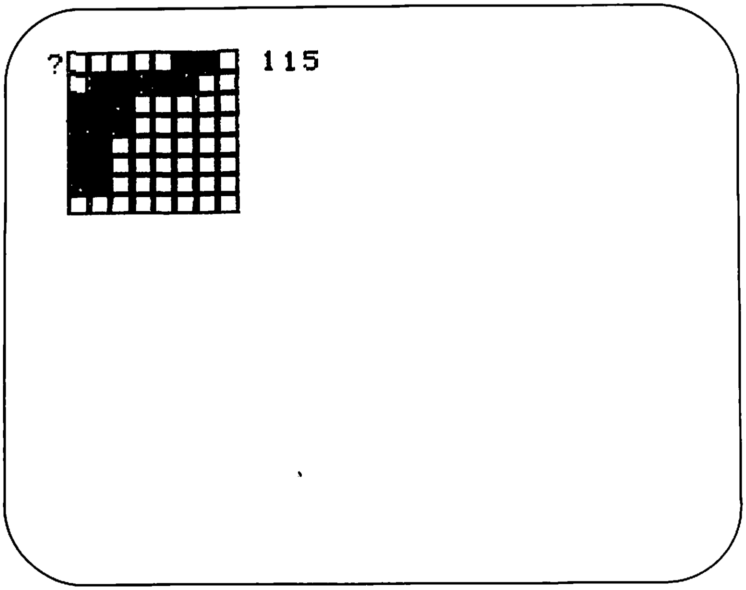

?
110

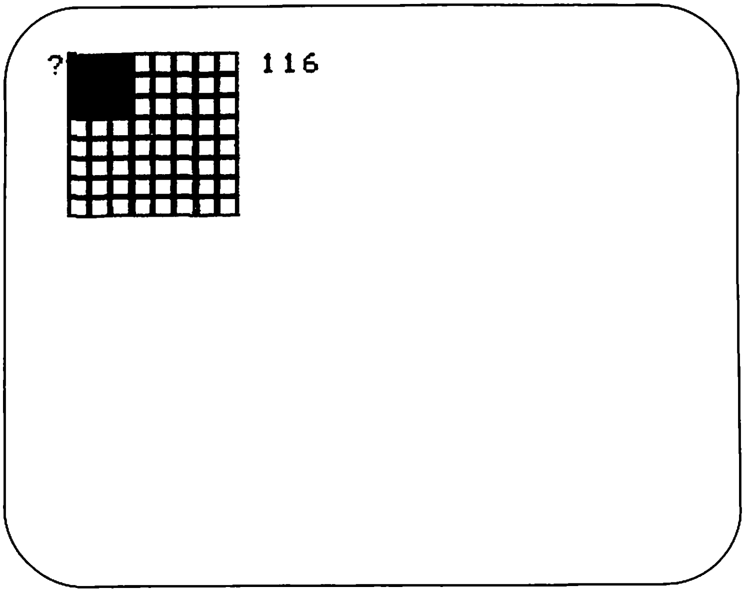

?
111


?
112


?
113


?
114

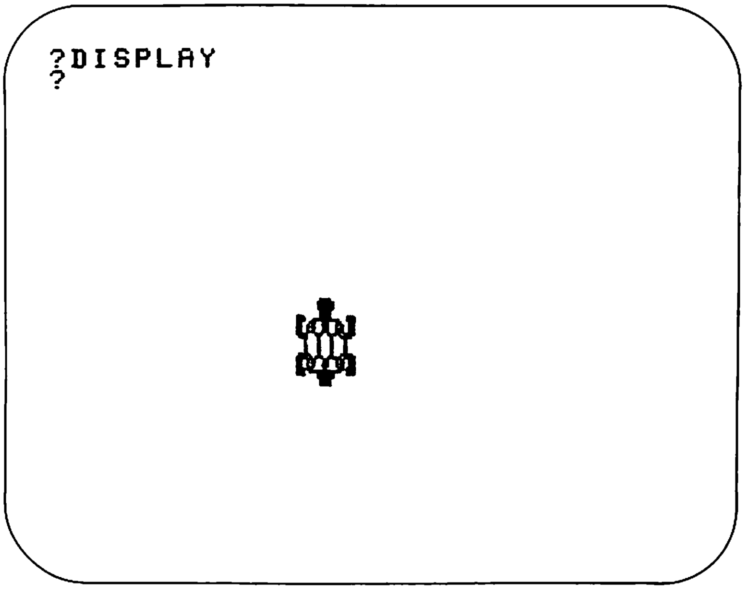

?
115


?
116

You might not be able to tell much about the final design by looking at these individual tile patterns, but when you enter

```
CS
DISPLAY
```

you will see that we have created a picture of a turtle!



Projects That Use Tiles

So far, we have explored two ways to use tiles as elements of pictures. The first way is to create a tile design that interlocks with itself in interesting ways when it is repeated. The second way is to create a picture from a mosaic of several tiles, each of which forms a separate and unique part of the total picture. One the powerful features of tiles is that they let you work with computer graphics on a dot-by-dot basis. This is quite different from turtle graphics, in which you tend to think more in terms of lines and curves.

Now it is time for you to create some tile projects of your own, such as creating a new wallpaper design.

Some people compare tile graphics to needlepoint or quilt designs; you might want to use such patterns as a guide for your own pictures.

Can you find tile patterns that produce interesting pictures when they are placed on the screen at random?

Since the alphabetic characters are tiles, you can create your own alphabet designs. Try designing an alphabet suitable for use in a love poem.

As you continue to experiment with tiles, compare your feelings about this form of computer graphics with your feelings about turtle graphics. Which form do you like better? Why?

In the next chapter, we will combine the dot-by-dot precision of the tiles with the turtle's sense of flow and motion in the creation of animated shapes called *sprites*.

X. Sprites and Animation

One feature of television is that it allows us to see moving images. Although your TI-99/4A computer uses a television screen for the display, all the figures we have examined thus far are static. In this chapter, we will explore ways that TI Logo lets you create moving images on your display.

TI Logo is equipped with a powerful graphics tool that lets us create shapes similar to tiles that can be placed anywhere on the screen (not just at character boundaries). Furthermore, these shapes (called *sprites*) can be assigned a velocity or speed with which they will move across the screen. By combining several sprites into a larger shape, multicolored moving objects can be created.

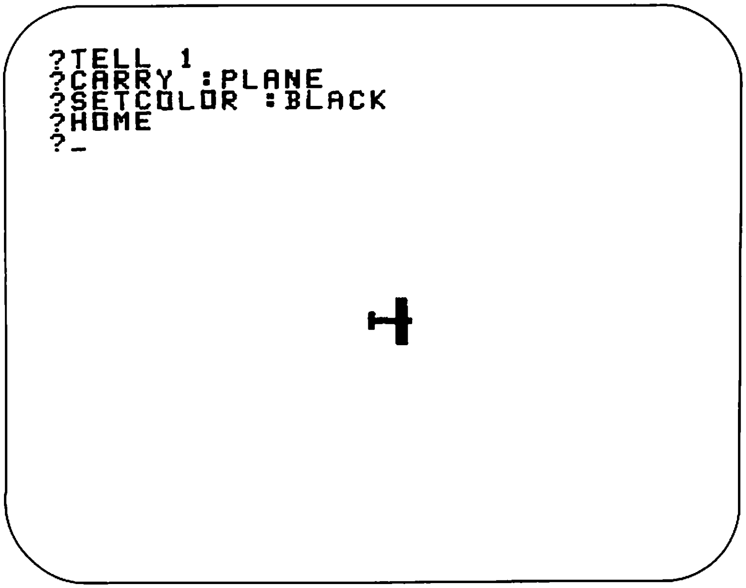
Sprites are the tools that allow you to create your own animated sequences. In a later chapter, you will learn how to record these sequences on videotape, and you will soon be able to create your own animated films from the comfort of your own home. First, however, we must learn how to create sprites and move them on the screen.

Moving Shapes

To see an example of a sprite, enter the following commands:

```
CS
TELL 1
CARRY :PLANE
SETCOLOR :BLACK
HOME
```

This sequence of commands should result in the appearance of an airplane on the display screen.



Here is what happened as you typed each line. TELL 1 lets Logo know that the instructions you type next are to be followed by sprite 1. There are 32 sprites, all of which can be on the screen at the same time (with some restrictions). If you are ever confused about which sprite (or sprites) you are in communication with, just enter

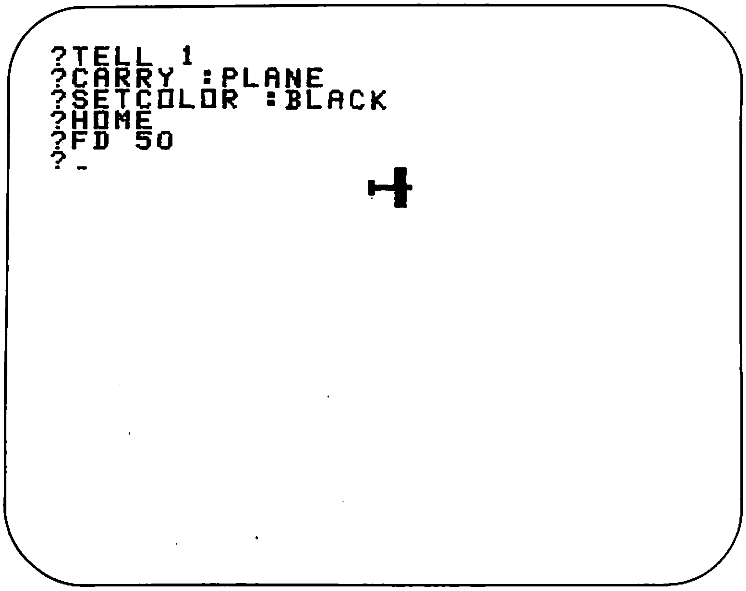
```
PRINT WHO
```

and the number(s) of the currently addressed sprite(s) will appear on the screen. The command CARRY :PLANE instructs the sprite to carry one of the predefined sprite shapes provided by Logo. The SETCOLOR and HOME commands are the same as the ones we used with turtle graphics. In fact, you will find much of your turtle graphics experience to be valuable as you study sprites.

Now that we have magically caused an airplane to appear at the center of the screen, let's see if we can make it move to another location. Enter

FD 50

This causes the sprite to move up the screen by 50 units, just as if it were a turtle.



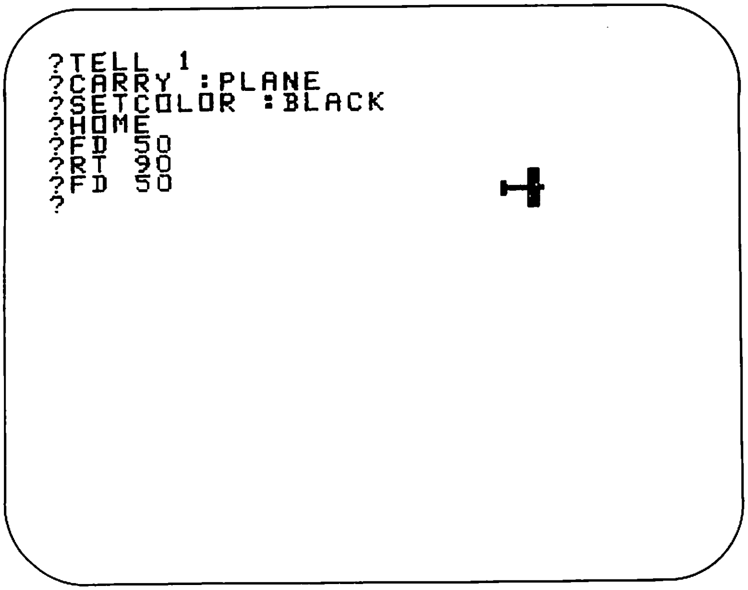
Now let's see what the other turtle commands do. Enter

RT 90

The image of the plane doesn't rotate when we enter this command, but if we now enter

FD 50

we can see that the plane has moved to the right by 50 units.



So far, we have found that sprites and turtles have some properties in common, but they have some differences as well. As you experiment, you will find that the location of a sprite can be changed with the familiar turtle commands, FORWARD, BACK, LEFT, and RIGHT. One difference between the turtle and sprites is that LEFT and RIGHT don't change the orientation of the sprite but do change the direction of the sprite's subsequent motion. When we told the sprite carrying the airplane to turn to the right by 90 degrees, the image of the plane didn't turn, even though it moved in the new direction when we typed FD 50.

Another difference between the turtle and sprites is that sprites don't draw lines; the messages PENUP and PENDOWN don't do anything. The SETCOLOR command, however, does do something—it sets the color of the sprite itself.

As you can see, sprites have a lot in common with the turtle, yet they have some unique properties of their own. To see another of these unique properties, enter

```
CS
```

(Note that the screen cleared, but the sprite remained visible.) Now enter

```
SETSPEED 20
```

The plane will start moving from left to right across the screen. You have imparted *motion* to a graphic object. When the plane reaches the right edge of the screen, it reappears automatically at the left edge—so you will never have to worry about losing a sprite. To temporarily freeze the motion of the plane, enter

```
FREEZE
```

To get the plane moving again at its original speed, enter

```
THAW
```

Thus, the SETSPEED (or SS) command imparts motion to sprites. This motion will be forward for speed values from 1 (very slow) to 127 (very fast) and backward for speed values from −1 to −127.

You can change the direction of a sprite while it is moving. As an example, enter

```
SS 30
RT 90
```

As soon as you entered RT 90, the plane changed direction; it is now traveling from the top of the screen to the bottom.

We can use this feature of TI Logo to send our airplane on a square path. The following procedure should do this very nicely:

```
TO PATH
SS 40
START:
WAIT 30
RT 90
GO "START
END
```

To try this procedure, enter

```
CS
HOME SH 0
PATH
```

The plane will now trace a square path on the right half of the display screen. When you tire of this path, press BACK to get out of the procedure.

Now that we know how to create a sprite, let's learn how to make it disappear. One way to do this is to use the following procedure:

```
TO VANISH :NUMBER
TELL :NUMBER
CARRY 0
SETCOLOR 0
SETSPEED 0
SETHEADING 0
END
```

If you now enter

```
VANISH 1
```

the sprite will vanish from the screen.

Making Your Own Shapes

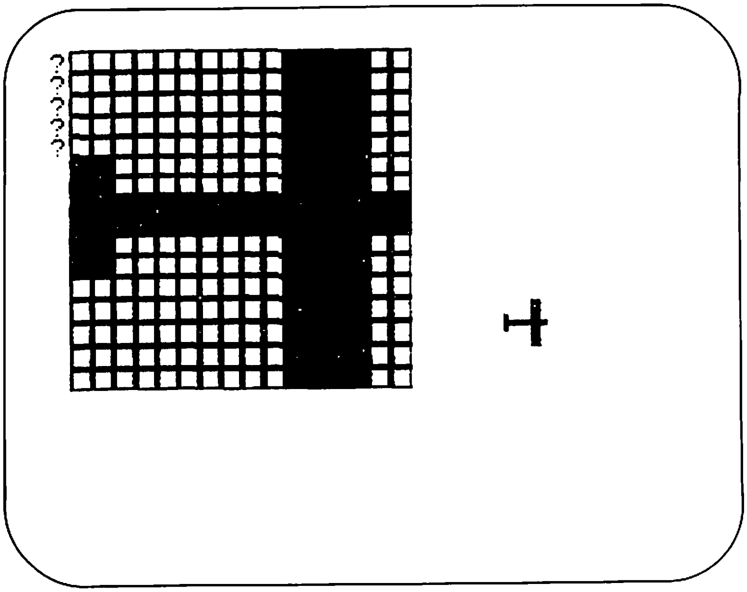
You may be surprised to find that TI Logo has already defined some sprite shapes for us to use. Just as with tiles, some sprite shapes have been defined and some have not. Of the 26 sprite shapes available to us, 5 have already been provided with patterns. These shapes are shown in the following table:

SHAPE NAME	SHAPE NUMBER
:PLANE	1
:TRUCK	2
:ROCKET	3
:BALL	4
:BOX	5

All the other shapes (0 and 6–25) are blank when the computer is first turned on. To define a shape of your own, you must use the MAKESHAPE command, which operates like MAKECHAR. As with character tiles, it is a good idea to display a sprite carrying the shape you are editing so that you can see both the enlarged and final-size shapes as you are making changes.

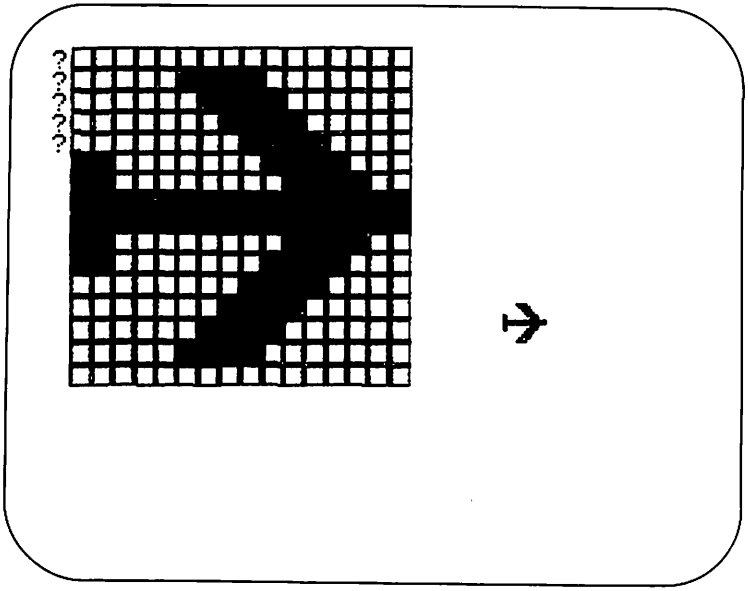
To illustrate the shape-defining process, let's change the shape of the airplane so that it has swept wings. Enter

```
TELL 1
CARRY 1
SC :BLACK
HOME RT 90 FD 50
MAKESHAPE 1
```



As you can see from the figure, large and small images of the plane both appear on the screen at the same time. If you count the boxes in the large grid, you will see that sprite shapes are made from larger matrices than those for tiles. Character tiles are formed from 8 × 8 dot arrays, whereas sprite shapes are formed from 16 × 16 dot arrays.

To change (or create) a shape, you use the arrow keys and the FCTN key in exactly the same way as you do for editing a tile. Using these keys, edit the image of the plane so that it has this shape:



When you have edited the shape to your satisfaction, press BACK to return to the normal display screen.

As with tiles, sprites can be saved on the diskette, along with any procedures you may have defined.

Now that our new jet plane has been created, you can send it streaming across the screen.

Many Sprites

You already know that many characters can be displayed on the screen at one time. The same is true of sprites. There are 32 sprites available for you to use. With a few restrictions, you can have all 32 on the screen at once. If you try to have more than eight of them on one horizontal line, they won't be displayed properly, but you can usually avoid this problem by careful sprite placement.

Each sprite can carry any of 26 shapes and can have its own color, speed, location, and orientation. TI Logo allows you to send messages to sprites individually or in any cluster you want. The ability to cluster sprites is powerful, because it lets you create multiple sprite objects that can be moved as a unit with a single command.

As an example, let's do an experiment with eight sprites. We can use any eight sprites we want, so let's pick sprites 0 to 7. First, we should reset all the sprites to their erased state. We can do this by typing

```
VANISH :ALL
```

since TI Logo has already defined the variable ALL to contain a list of all sprites.

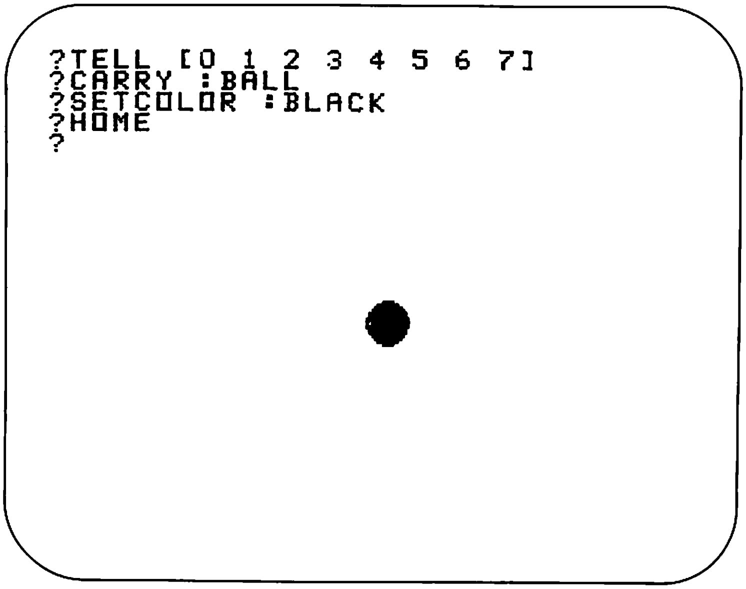
Next, enter

```
CS
TELL [0 1 2 3 4 5 6 7]
```

This sets up eight sprites so that all subsequent messages will be obeyed by the group as a whole. Thus, for example, if we enter

```
CARRY :BALL
SETCOLOR :BLACK
HOME
```

we will see a single ball in the center of the screen. In fact, all eight sprites are there, carrying identical shapes with identical colors, and they are stacked one on top of the other.



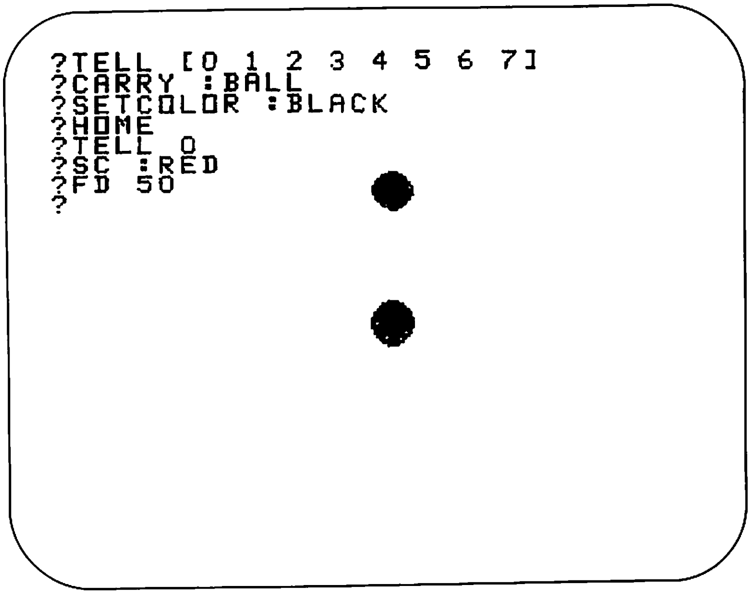
What is the stacking order? To find out, let's send some messages to the sprites one at a time. Enter

```
TELL 0
SC :RED
```

The ball turned red as soon as we pressed ENTER, showing that sprite 0 is on the top of the stack. If you now enter

```
FD 50
```

the movement of the red sprite reveals the black sprite (number 1) underneath.



If we send the message

```
TELL 1
RT 90
FD 50
```

we will move sprite 1 to the right and reveal sprite 2. We can establish the identity of sprite 2 by entering

```
TELL 2
SC :GREEN
```

Now we can create all kinds of havoc on the screen by sending each sprite off in a different direction and at a different speed. As an example, enter

```
TELL 0
SS 25
TELL 1
SS 50
TELL 2
LT 30 SS 30
TELL 3
RT 30 SS 10
```

and so on. You will see a dazzling array of moving balls on the screen. To stop the motion, enter

```
FREEZE
```

and all the sprites will stop. Enter

```
THAW
```

and they will all start moving again. To make all the sprites disappear, enter

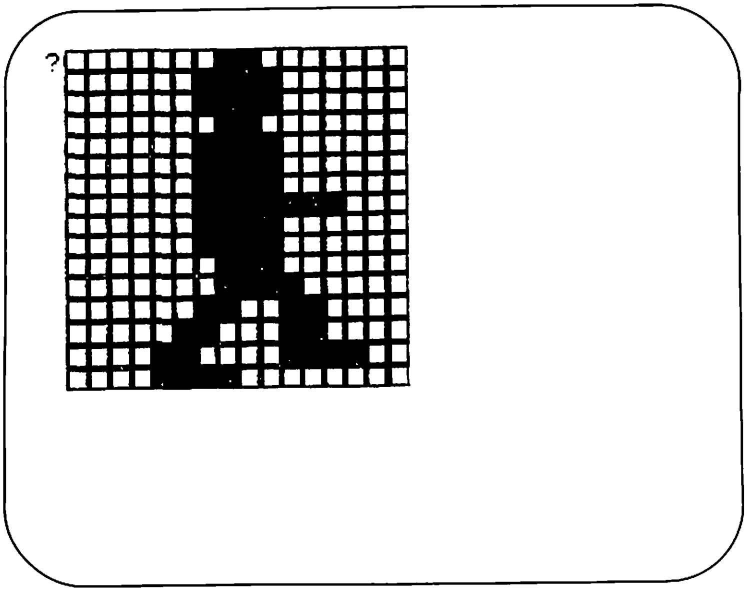
```
VANISH :ALL
```

One Sprite, Many Shapes

The ability to make a shape move across the screen is only one aspect of animation. To see why we need even more techniques, let's experiment with the shape of a human figure. Create the image as shape 6 by entering

```
MAKESHAPE 6
```

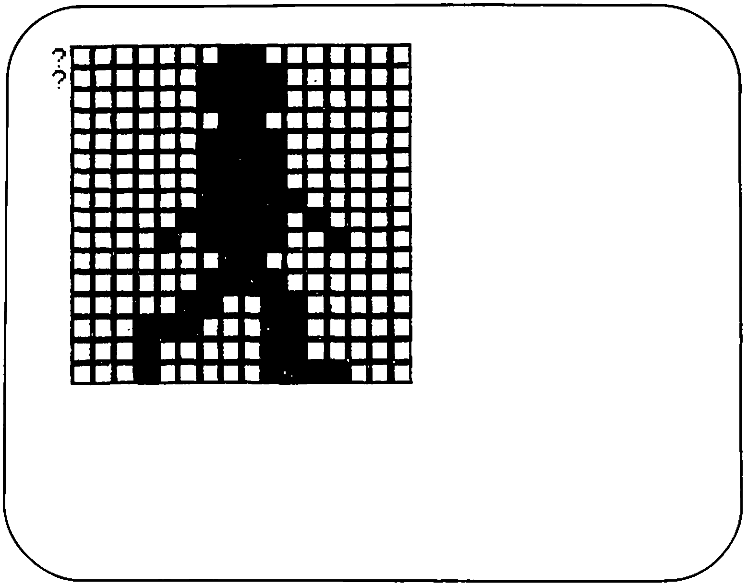
and edit it to look like this:

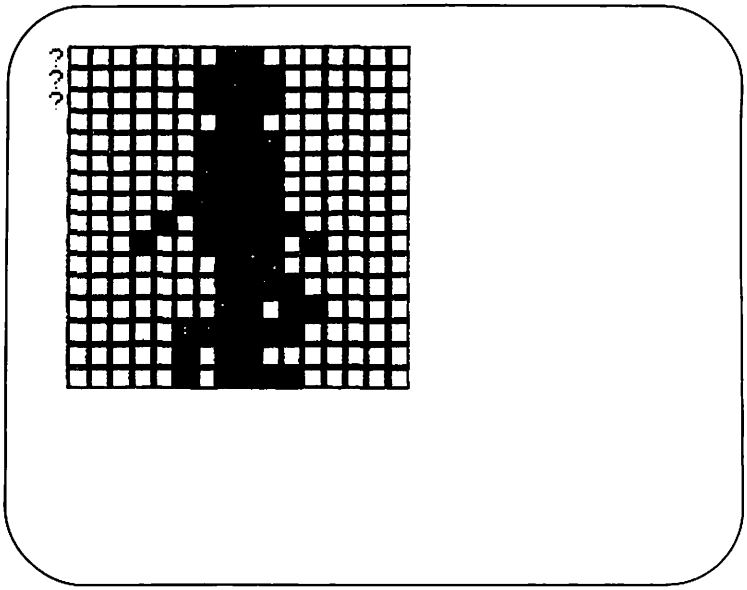


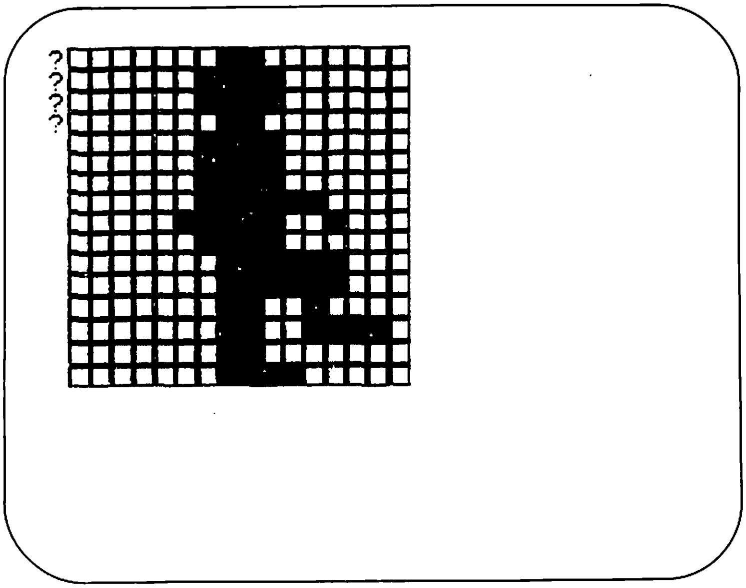
If we now set this figure on the screen and make it "walk," we will see that something is very wrong. Enter

```
TELL 0
CARRY 6
SC :BACK
HOME
RT 90
SS 15
```

The figure moves from left to right across the screen, but it appears to be floating rather than walking. To fix this, we need to define some more shapes that show intermediate body positions while it is walking. The next few shapes should help us a great deal. Define these in order as shapes 7, 8, and 9:







Finally, to get the illusion of walking, we not only need to have the sprite move, but we need to repeat the sequence of walking frames over and over again. The following procedure will let us experiment with this:

```
TO WALK :SPEED :GAIT
TELL 0
HOME SC :BLACK
SH 90 SS :SPEED
START:
CARRY 6
WAIT :GAIT
CARRY 7
WAIT :GAIT
CARRY 8
WAIT :GAIT
CARRY 9
WAIT :GAIT
GO "START
END
```

If you now enter

```
CS
WALK 7 5
```

you should see a far more realistic image of a person walking across the screen than you did at first. Experiment with different speeds and gaits to find other combinations that look proper.

If you are interested in making your images look even more realistic, you should read such books as *The Animation Book*, by Kit Laybourne (Crown Publishers). Although it is oriented to film animators, this book also offers much of value to computer animators.

Projects that Use Sprites

If you have ever tried to create animated sequences using other techniques, you can appreciate what a time-saver sprites and computer graphics can be. Our project of animating a walking figure, for example, would not be a normal starting point if we were experimenting with frame-by-frame animation techniques.

A more common starting point for experimenting with animation is to create a sequence that shows a bouncing ball. Using the techniques covered in this chapter, write a procedure that has a ball bounce near the bottom of the screen and go back up again. Next, define a shape for a ball with a slightly squashed bottom, and modify the procedure so that when the ball reaches the bottom, it looks compressed before it goes back up.

Since the normal cyan background color makes a nice sky, experiment with the use of sprites to make some fluffy white clouds. If you start out with four or five overlapping cloud-shaped sprites and have them move apart from each other, you can create changing weather patterns. If you slowly spread the clouds out and then change them from white to gray, you can create the beginnings of a thunderstorm.

Using one sprite for a tree trunk and another to hold the shapes for the leaves, create an animated sequence to depict the changing of the seasons. In the spring, the bare trunk should sprout leaves that become full by summer. In the fall, the leaves should change from green to rust and then fall off. The tree should be covered with snow during winter, and the snow should melt as spring approaches.

As you gain facility with sprites, you will want to intermix sprites with character-based (or turtle-based) background images. By combining graphic techniques, you will discover that you have all the tools you need to create animated stories on your television screen. Your patience can pay handsome dividends—especially when you realize how much easier it is to create animated sequences using computer graphics than it is using traditional animation techniques.

XI. The Creation of Animated Sequences

The preceding chapter covered the last of the graphic techniques available to us with TI Logo. Rather than bringing us to the end of our studies, however, it has finally allowed us to begin! You see, the real task is not just to learn the techniques of computer graphics—that is easy enough to do. The real task is to apply these techniques in the creation of your own artwork.

The creation of static works of art has been covered in the preceding chapters. The creation of animated artwork is sufficiently complex, however, that we will devote this entire chapter to it.

At best, this chapter will serve as a set of hints that might be useful to you. Some of these hints will be more helpful than others. As with any creative endeavor, however, you must find your own path—your own method of design and expression. If you have worked with animation before, much of what you already know will give you a head start. If you haven't worked with film or flip books, your freshness to the subject of moving pictures may provide you with insights that might otherwise be missed.

Animation can take many forms, ranging from the Saturday morning cartoons to such abstract masterpieces as John Whitney's *Arabesque*.

Whether your sequences tell a simple story or are designed to evoke a purely emotional response, all animated sequences have a design with a beginning, a middle, and an end. If you have a clear idea where your artwork is heading, the result will be successful, no matter how simple its execution might appear. If your sequence rambles or is aimless, your efforts to communicate will fail, no matter how sophisticated your images may be. No amount of experience or technique can compensate for the lack of flow or continuity in a sequence—and that flow and continuity is completely in your hands.

Because of the importance of knowing where you are going with a piece, we will spend some time exploring a major tool of the animator—the storyboard.

Storyboards

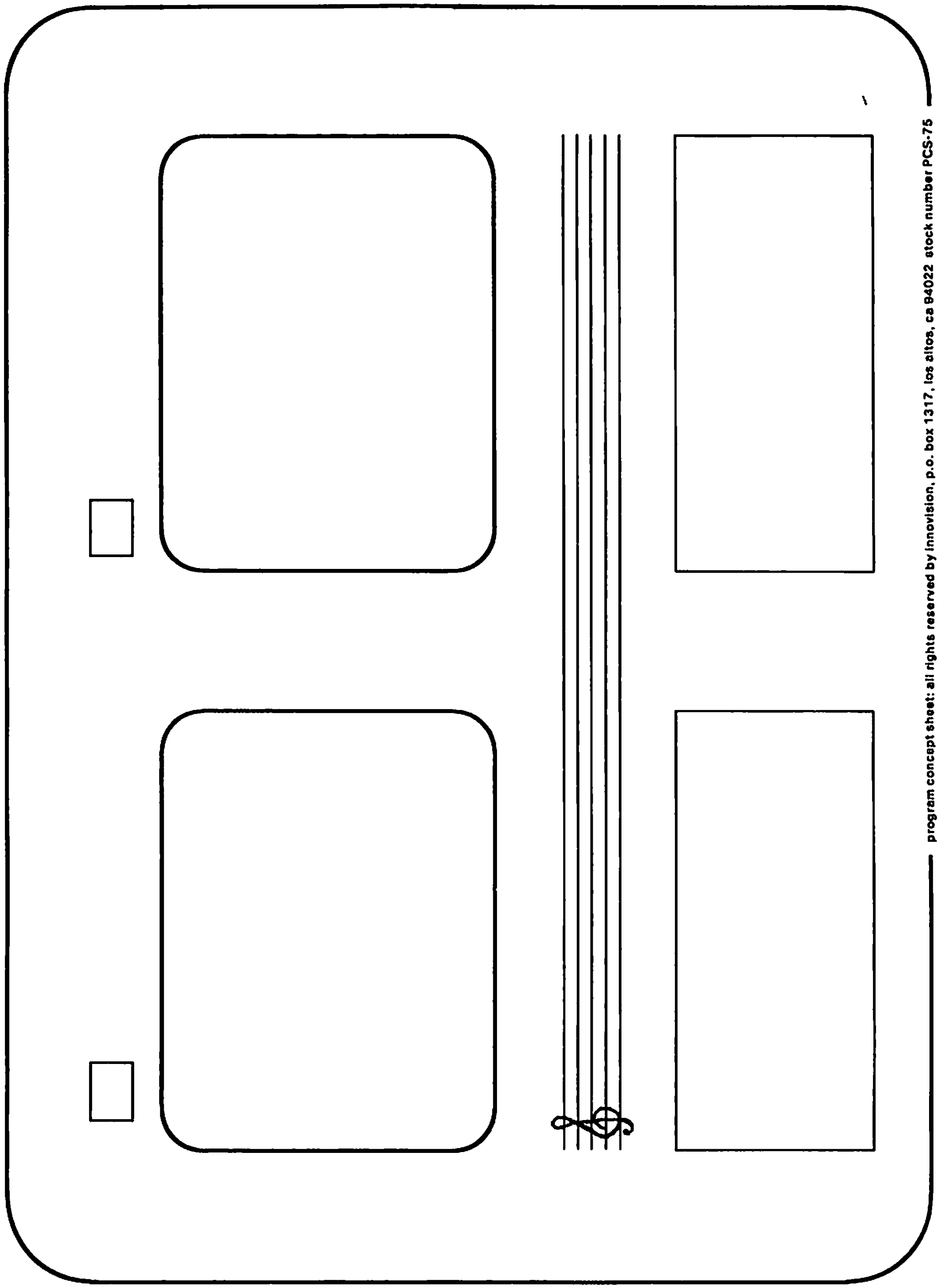
Of all the tools of value to the animator, the simplest and most valuable is the storyboard. A storyboard is a conceptualizing tool that lets you examine an animation project scene by scene. To make a storyboard, you create a series of small sketches of key frames and tack them up on a wall, where the project can be looked at as a whole. Each element of a storyboard is a sketch of one frame. These sketches may be quite simple, showing only the general background and figure placement. The goal is not to make a perfect model of the final frame but to do a quick rendering that pins the key concept for later reference.

How many frames you should sketch depends on the project. Generally, you will find that you will want to see the starting and ending positions of each motion sequence. The storyboard can help you make smooth transitions between scenes and will serve as the outline for your piece.

Although just about any bits of paper can be used for your storyboard, you might want to copy the format shown on page 195. This format has room for a starting and an ending frame in the curved boxes, a musical staff for notating sound effects or other musical additions, and two rectangular boxes at the bottom in which you can write descriptions of the action you want to create or in which you can make other notes to yourself.

Depending on the length of your story, you may have a few key frames or you may have over a hundred. No matter how many you have, it is important to complete the storyboard before beginning the task of creating the final animation sequence. Remember that the storyboard is a practical, not an aesthetic, tool, so don't labor over your sketches.

When you have tacked your images on the wall, look at the result carefully. Do the sequences lead smoothly into each


program concept sheet: all rights reserved by innovision, p.o. box 1317, los altos, ca 94022 stock number PCS-75

other? Is there a sequence that requires a special technique you should master first? The overall flow of your story should be apparent from your storyboard. If potential viewers can't figure out what you are doing from the storyboard, they probably won't be able to understand the finished project either.

Although the storyboard can help you present your ideas to others, it is primarily a personal tool to help you to clarify your ideas. Good storyboarding is the foundation of good animation and is thus deserving of some practice on your part.

Starting Projects

A common tendency in choosing an animation project is to pick one that is too complex. No matter how skilled you are as an artist or computer programmer, you won't be tackling a project like Lisberger's *TRON* as your first animated artwork.

Probably the best starting point is for you to do some "studies." These studies should be short animated sequences (usually 10 seconds or less) that will help you create and refine your graphic techniques. The bouncing ball experiment in the last chapter is a good example of a study.

What kinds of projects make good studies? Consider the walking figure sequence we created in the last chapter. We made the figure look more realistic in its motion by going from one shape to a set of four shapes repeated in sequence. Even so, our figure's gait was somewhat stilted. If you analyze the motion carefully, you will find that the head and torso of the figure don't move up and down at all. You might want to spend some time studying walking. Can you make the figure walk on its toes? Can it walk in a slouch? Can it skip like a child? You might have the character go from a standstill to a brisk walk in gradual steps, rather than going immediately to its final speed. Learn how to make the figure walk partway across the screen and stop before going on.

There is no limit to the number of animation sequences that can serve as excellent studies—the bounce of a ball, the splash of a raindrop, the movement of a cloud, the setting of the sun. Each of these can be a powerful learning experience and source of great pleasure to you.

As you create your studies, keep good records of your procedures and sprite shapes for future use. For the studies, you needn't concern yourself with background illustrations or painstaking detail. You are learning how to analyze and model motion, and that is a large enough task at this point to consume all your effort.

Creating Short Scenes

When you have gained some experience with specific animated sequences of interest to you, it is time to animate some complete scenes. These scenes should also be short—perhaps as long as 30 seconds and certainly no longer than a minute. The goal here is to tell a simple story. You should design a background scene using either tile or turtle graphics. This scene will remain stationary, and all the sprites will appear to move in front of it. An alternative approach, commonly used to convey the feeling of motion over long distances, is to have a "moving" object of interest, such as a truck, remain stationary and to have the immediate background (for example, houses and trees along the roadway) move in an opposite direction to the truck's perceived motion. This creates an excellent illusion of motion and keeps the object of focus at the center of the screen. You may find that this technique is a little trickier to implement than scenes with stationary backgrounds and thus may want to defer it to a later experiment.

As with the studies, strive for simplicity. You will be learning how to integrate your sprites with a background. The background shouldn't be too busy, or attention will be diverted from the animated portion of the scene. Only your own taste and experience can guide you effectively here.

As a guideline, the scene should be sufficiently simple that it all fits in the computer's memory at the same time. The scene should also have a simple theme. An abstract sequence might show the metamorphosis of one geometrical shape into another; or you might want to depict a scene from a haiku. Your story should be clear enough to be understood by anyone who will see it. Try to depict simple events—driving through the woods or walking on a beach with sea gulls flying overhead.

You might want to depict an event of history—such as the apple falling on Newton's head—but don't try something as ambitious as the Battle of Hastings!

In designing your sequence, make heavy use of storyboards. Once the scene is clear in your mind, you are ready to put it into the computer. Generate the background procedures first. If you use tile patterns, try to develop a few generic tile shapes that can be used over and over again. You can then build the background much as you would construct something with bricks. Remember that the background should complement and support the action that takes place in front of it. Keep the background simple. If it is too busy, the animation will fail. Choose colors carefully; keep the background in neutral tones and make sure it isn't distracting.

When the background is finished, save your procedures and tiles on a diskette. In fact, you should be saving your files every 30 minutes or so, just in case the power goes out or you accidentally push QUIT.

To develop the animated portion of the sequence, you might want to make a first attempt at creating all the sprite shapes you will need. Don't worry if they aren't perfect; you will be able to edit them later. Make notes on your storyboard, assigning a number to each shape as it is created. This will help you later when you go to put everything together.

Next, start writing the animation procedures that will place the sprites on the screen and set them in motion through the action of the scene. Again, this can be done roughly at first. If you know where you want the sprites to go but aren't sure of the speeds, make the speeds into variables that can be specified when the procedure is run.

Finally, use the background procedure to draw its image, and then run the sprite procedures to see how they work. Make note of shapes and motions that seem awkward. Is the motion too sweeping or too stilted? Are the sprites the right shape, size, and color? If one object is shown in several positions (such as a walking figure), does each pattern lead smoothly to the next?

Refine your sequence until it is perfect, and then make a copy of it on your diskette. You should use a separate diskette that becomes your portfolio of finished works.

In viewing your completed work, you might feel that it would be enhanced by music or narration. In the next chapter, we will show how to add this accompaniment to your recorded performances.

Music can be a very powerful addition to your animated works. As you listen to music, think about how the mood, color, and tempo fit with your scenes. You will be surprised at how easily music blends with computer animation. I tend to prefer simple rather than heavy music to accompany abstract animation sequences. In particular, I find some works by Debussy, Satie, Ravel, and Varese quite appropriate. You should choose your own favorites, however—jazz, classical, rock, whatever you want. Remember that you are expressing your own ideas, not someone else's. To be honest, your artwork should reflect *your* tastes.

Creating Longer Works

When you have made several studies and have completed a few scenes, you may be ready to tackle a complete story. Again, think small. These projects take a lot of time, and by keeping them tractable you will enjoy your learning experience rather than feeling burdened by it. An animation project with a running time of 5 minutes can be a tremendous effort, but you should be ready for it! As with all projects, plan exactly what you want to do and create a storyboard. Story topics might range from folk tales to an abstract choreography to a piece of music. Remember that all your programming tools are available to you—turtle graphics, tiles, and sprites. If you are creating a piece that is to fit with a specific piece of music, consider using the joystick to start procedures at exactly the right moment, rather than trying to time everything to run perfectly from scratch.

When you have picked the theme for your piece, create the storyboard and see how the result feels to you. Do you have too many scene changes or not enough changes? Can one background be used in several scenes? How many sprite shapes will you need? How will your layout make a smooth transition from scene to scene? (The technical details of this step will be covered in the next chapter.) As you answer these questions, separate your storyboard into blocks representing specific scenes. Each of these scenes should be complete in itself and probably should be stored as a separate file on the diskette.

Above all, remember that even though your animation project is separated into chunks, these chunks must all flow together as one coherent whole to make the project work. This integration task is very important. You may find yourself spending a great deal of time "tweaking" the end of one scene to make it flow smoothly into the next.

If your project is sufficiently complex, you may find that it will have to be interrupted at the end of each scene to change procedures. The only way to do this smoothly is to record each scene on a video recorder, one at a time, and then play the completed work out as a whole. The next chapter will show you how to do this.

XII. Artwork Recording Techniques

Although the creation of graphic images and animated sequences may provide a great deal of personal satisfaction, you probably would also like to share the results of your effort with others. Clearly, your computer system is too cumbersome to carry around to show to all your friends, so a more transportable form for your artwork must be found. This chapter describes two ways of capturing your artwork for later viewing—the use of a camera for recording still pictures and the use of a video cassette recorder for recording animated sequences.

Recording Static Images

There are several ways to make color photographs of your computer-generated artwork. The easiest technique is to photograph the image directly from the television screen with a camera. Although many types of cameras can be used effectively, I prefer the use of a 35 mm single lens reflex camera, so that I can see the image exactly as it will appear on the film.

To photograph the screen, you will have to use an exposure of 1/30 second or longer. The reason for this is that the image you see on your television screen is formed from two interlaced scanned patterns, each of which takes 1/60 second to form. It takes two scans to build the whole image, so a shutter speed of 1/30 second is the absolute minimum for capturing the whole image. Depending on the brightness of your television screen and the speed of the film you are using, you may find that even longer exposure times are required. Because of the length of the exposure times involved, you must have your camera mounted on a stable tripod to prevent blurring of the images.

The tripod height should be adjusted so that the camera lens is perfectly centered on the screen. By centering the lens in this way, you eliminate an undesirable effect called keystoning. Keystoning occurs when an image is photographed or projected at an angle. The result is that one edge of the image appears larger than the opposite edge. A square photographed this way will take on the trapezoidal shape of a keystone—hence the name. So remember, if you want your pictures to be as accurate as possible, you will have to adjust your tripod height accurately.

When you make your photographs, you will be capturing everything that appears on the television. This includes the image you want as well as finger prints on the screen and any reflections from other light sources. Before photographing any images, clean your television screen with a damp cloth and a mild soap. If the TV hasn't been cleaned in a long time, you may be surprised at the accumulation of dirt. Because of the high voltages used in televisions, the display screen usually acquires a static charge that acts like a dust magnet.

When you are ready to start photographing your images, the only source of light in the room should come from the display screen itself. I find that I get my best results at night in a completely dark room. If adjacent rooms are illuminated, you can seal light leaks around doors with masking tape. Also, if your television has an illuminated channel selector, block its light with tape.

The choice of film type is largely up to you. For color prints, films such as ASA 100 Kodacolor II give fine results. For slides, you may wish to use a high-speed Ektachrome (ASA 400) or a slower-speed Kodachrome. The main difference you will notice between these films is that Kodachrome tends to produce brighter and warmer colors than Ektachrome. Experiment with film types to see which you like best.

The distance between the camera and the TV should be chosen so that the computer image perfectly fills the viewfinder. If the image is too small, move the camera forward; if it is too large, move the camera back. As you adjust the focus on the camera, the image may get smaller or larger. When the focused image is the right size, you might want to mark the tripod location on the floor with masking tape to simplify setup the next time you are ready to take photographs.

The next step in your picture-taking experience is to adjust the lens opening for the correct exposure. You should never rely on the light meter built into your camera when photographing a screen image. You will want to control both the shutter speed and lens aperture controls manually. Your best bet is to use a separate light meter and take your readings from the front of the TV screen. You should take readings from several areas of the screen. When you have made your readings, follow the instructions on your light meter for adjusting your camera. If you can't find an appropriate lens opening for a shutter speed of 1/30 second, set the shutter to a slower speed (for example, 1/10 second) and adjust the lens opening to the correct value. You will find that aperture settings in the range of f4.0 or higher will provide you with some tolerance in focusing. If you have your lens set at an opening below f2.0, be sure that your camera is well focused before taking any pictures.

The first time you photograph a series of images, you should repeat the same shot several times with different lens openings. Keep a log of your settings, so that when the film is developed, you will know the optimal settings for your camera.

Although the foregoing technique is perfect if you already have a camera, it is rather cumbersome—especially if you are going to take hundreds of pictures. There are special cameras designed just for taking pictures of TV images. These professional camera systems are fairly expensive (typically, more than $2000), but they produce results far superior to images photographed directly from the TV screen.

I make color photographs with an Image Resources Videoprint 5000 system. Other camera systems, such as those made by Lang Systems, Inc., operate in similar fashion. Rather than displaying the image on a color television tube, these systems electronically separate the display signal into its red, green, and blue components. Each signal is then displayed on a black-and-white monitor, one at a time. A color filter wheel is placed between the display screen and the camera lens. When the red image is displayed on the black-and-white screen, the image is recorded on the film after passing through the red filter. This process is repeated for the green and blue filters. By exposing each component separately, the full color image is reconstructed on the film. Since color filters can match the primary colors far more accurately than can the phosphors used in color televisions, the result is a photograph with far greater brilliance and depth than can be obtained otherwise.

You would want to be quite sure of your serious interest in computer graphics before investing in such a system, but the results easily justify the cost for the serious graphic artist.

Once your images are on film, you can continue to be creative. Let's suppose, for example, that you have made an 8″ × 10″ black-and-white print of a turtle graphic image of a squiral. If you have access to a thermal copier, your local art supply store might be able to supply you with a heat-sensitive medium that would allow you to transfer the image to a silk screen for about a dollar. Once this screen is made, you can print the image, using traditional screening techniques, in any color and you can transfer it to paper, wood, or fabric. You can even make your own computer graphic T-shirts.

As with the design of the images themselves, you should let your imagination run free when you think about things to do with the photographed copies of your computer-generated artwork.

Recording Moving Images

Besides the television set itself, one of the most popular entertainment products seems to be the video cassette recorder (VCR). Normally, a VCR is used for playing back prerecorded movies or for recording programs that you would like to view at another time. By connecting your computer system to a VCR, however, not only can you capture your images on tape for others to see, but you can string several short animated sequences together to create a complete animated film.

There are many advantages to using a VCR to record your video masterpieces. First, you can see the results immediately and can re-record sequences that don't look right. Second, because of the low cost of videotape (compared to film), you can record hours of computer graphics inexpensively. Also, unlike using a camera to photograph the screen, the VCR can be used in broad daylight. If you don't yet own a VCR, you might want to borrow one from a friend to see if you like using it.

Two major videotape formats are commonly used in the home—Beta and VHS. Because most video recorders have the same overall features, it doesn't matter which format you use. Be aware, however, that these tapes are not interchangeable. You can't play a VHS tape on a Beta machine, and vice versa.

If you are shopping for a VCR of your own (some models are fairly inexpensive), here are some features to look for. To record multiple sequences of animation, you will need a recorder with a PAUSE feature. The PAUSE button immediately stops the tape to allow you to load a new sequence into the computer. When you play the tape back, your sequences will appear with no time gaps between them.

Another feature to look for is direct video input and output jacks. These let you record from your computer directly, without using a radio frequency (RF) modulator (the box you connected to your TV set). Most people feel that direct video input produces a higher-quality image. To use this feature, you will also have to buy a video monitor cable for your TI computer, which is available from your dealer. Aside from improved video quality, the use of direct video input lets you use some special-effects equipment to enhance your images.

There are other features that your VCR might have. The ability to freeze frames and make images move in slow motion can be very valuable in letting you analyze your work in detail. VCRs are available in many price ranges, and you should select one that has the performance characteristics you want but stays within your budget.

When you have acquired your VCR (and some blank tape), you are ready to connect it to your computer. The easiest way to do this is to connect the TI RF modulator to the VCR antenna terminals, as shown in the manual that came with your video recorder. Your television set should also be connected to the VCR, so that you can see the images as they are being recorded.

Now that everything is hooked up, you are ready to start recording your images. Rather than starting with an animated sequence, we will start with a "slide show" of your static artwork. By now, you probably have quite a few pictures saved on your diskette. To record these images on the videotape recorder, you might first want to make a procedure that displays a title page.

You can be quite imaginative in the creation of title pages—having the title move across the screen character by character or changing colors of characters after they are displayed. The easiest way to make a title screen would be to use a procedure similar to the following (but with your own title and name instead of the ones shown):

```
TO TITLE
NOTURTLE
CS
PRINT [ ]
PRINT [ ]
PRINT [ ]
PRINT [REFLECTIONS ON A SILVER TREE]
PRINT [ ]
PRINT [A COLLECTION OF COMPUTER]
PRINT [GRAPHIC IMAGES BY]
PRINT [ ]
PRINT [AMY DOAKES]
PRINT [ ]
PRINT [1983]
END
```

Starting with this title procedure, design even better ones for your own use. You might also want to create a procedure called THEEND to provide more credits at the end of the show.

Let's suppose that you have several pictures in your computer memory at the same time. For simplicity, we will assume that these are called PICT1, PICT2, and so on. If you want to record each picture as it is being drawn, you should first make a procedure called NEXT that lets you advance to the next picture by pressing any key:

```
TO NEXT
MAKE "KEY RC
END
```

Load the tape in the VCR, make sure it is rewound, and then enter

```
TO SHOW
TITLE NEXT
PICT1 NEXT
PICT2 NEXT
.
.
.
THEEND NEXT
END
```

If you now enter

```
SHOW
```

the screen will clear and you will see your title image on the TV screen. At this point, press RECORD and PLAY on your VCR (or whichever combination of buttons is needed to start the recording process for your machine). After enough time has elapsed to be sure everyone can read the title page, press any key on the computer to advance to the next picture. Keep each picture on the screen long enough for everyone to look at it before advancing to the next image. Keep repeating this process until you have come to the end of your show.

If you have more pictures to add to the tape, press PAUSE on the VCR and load the new pictures into the computer. When you are ready to start again, press the PAUSE button again and you can pick up where you left off.

After completing your first recording, rewind the tape and play it back. Pay attention to details. Are the pictures on the screen for the right amount of time? Are they on too long? Is there one picture that isn't shown long enough? Are the pictures in the correct sequence? Try recording the same set of pictures in a different sequence to see if the result is better or worse. Remember that the advantage of the VCR is instant playback; use this feature to help you perfect your technique.

After you have successfully recorded some static images, you should record some of your animated sequences. Start by taking some of your studies and recording them with a title screen at the front and a closing text screen. After you are comfortable with your recording expertise, you are ready to make a tape of a longer animation project using several scenes—a movie!

Before you start to record your movie, make a note of which scenes you are using and which files they are on. Start with the title image and record scene 1. As soon as the action finishes, press PAUSE on the VCR and load your next scene. (Remember that you probably will have to erase the old scene to make room for the new one.) When the screen is loaded, start it and release the PAUSE button. Repeat this process until all the scenes are recorded, and then record the closing text frame.

When you are convinced that your masterpiece is visually complete, you may be able to add an audio track to your tape if your recorder has an overdub control. This feature allows you to add music or voice to your recorded tapes through a separate audio input jack on your VCR. Because of differences among models of video equipment, you should check your VCR owner's manual to see just how to use this feature.

At last you are truly on your own! You have mastered computer graphics and animation and have learned how to record your artwork on film and videotape for others to see. As you gain more expertise, you will find that you can use other video accessories to enhance your artwork. Special video equipment, such as titling machines, is now becoming affordable to home video enthusiasts. These machines let you mix video from a camera with your computer graphic images. Other video accessories let you enhance your images, change colors in subtle ways, and perform other feats of video magic.

Although you have come to the end of this book, you have only begun to explore the world of computer graphics and animation with TI Logo. As you continue to gain experience and expertise with computer graphics, you will find that this medium opens new worlds for you. If you are new to computers, I hope you have found this medium to be interesting. If you are new to graphic art, I hope you have learned to tap those creative resources buried inside of you.

Above all, have fun!

Index

> $12.95 FPT USA

David D. Thornburg

Computer Art and Animation

A User's Guide to

TI-99/4A Color Logo

Turn Your Computer into an Electronic Easel!

Computer graphics has too long been thought of as the domain of experts—complicated and very expensive. But now, owners of the inexpensive TI-99/4A computer and the powerful but easy-to-understand programming language Logo can learn to use their computers to create attractive, professional-looking graphics.

Whether you're an artist, a student, or simply an interested home programmer, here is a book to get you started on using your computer for creative expression. **COMPUTER ART AND ANIMATION: A USER'S GUIDE TO TI-99/4A COLOR LOGO** will take you on an artistic and programming adventure that will:

- Give you an easy-to-understand introduction to TI Logo that even a first-time computer user can follow
- Show you how to draw simple geometric shapes that become the basis for complex drawings and designs, including how to use "tiles"—the redefined shapes of characters already built into your computer
- Explain how to use the keyboard and joysticks to give you greater flexibility and efficiency in drawing
- Demystify sprites and animation, to enable you to create your own cartoons

From designing games to building an electronic artist's portfolio, **COMPUTER ART AND ANIMATION: A USER'S GUIDE TO TI-99/4A COLOR LOGO** will guide you through the use of a unique set of tools essential to the modern artist.

David D. Thornburg is a well-known educator and enthusiastic advocate of a humanistic computer revolution. He is the author of several books, including **PICTURE THIS!, PICTURE THIS TOO!,** and **DISCOVERING APPLE LOGO.**

ISBN 0-201-07958-5

ADDISON-WESLEY PUBLISHING COMPANY