dilithium Press

CRYSTAL CLEAR

LOGO (and more!) for the Commodore® 64™



LOGO (and more!) for the Commodore® 64™

LOGO (and more!) for the Commodore® 64™

By Richard Lambert

Illustrated by Anita Jones



dilithium Press Beaverton, Oregon

© 1984 by dilithium Press. All rights reserved.

No part of this book may be reproduced in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system without permission in writing from the publisher, with the following exceptions: any material may be copied or transcribed for the nonprofit use of the purchaser, and material (not to exceed 300 words and one figure) may be quoted in published reviews of this book.

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging in Publication Data

Lambert, Richard, 1947–
Logo and more for the Commodore 64.

Includes index.
Summary: An introduction to the LOGO computer language, with instructions for writing programs to draw pictures, write music, and other activities.
1. Commodore 64 (Computer — Programming — Juvenile literature. [1. Commodore 64 (Computer) — Programming. 2. LOGO (Computer program language) 3. Programming (Computers) 4. Computers] I. Title.
QA76.8.C64L35 1984 001.64'24 84-9620
ISBN 0-88056-348-6 (pbk.)

Commodore, Commodore 64, VIC 1541, VIC 1525 and Easy Script are registered trademarks of Commodore Business Machines, Incorporated.

Logo is a registered trademark of M.I.T. and Krell.

Terrapin is a registered trademark of Terrapin, Incorporated.

dilithium Press is a trademark of dilithium Press, Limited.

dilithium Press
8285 S.W. Nimbus
Suite 151
Beaverton, Oregon 97005-6401



Acknowledgements

I wish to thank all of my friends who gave me the support and help I needed to write this book, especially Janice who didn't believe I would really do it.

I would also like to thank Christopher McCord for testing the book.

Table of Contents

Introduction 1

Chapter 1
Getting Started 5
System Requirements 5
The Commodore Keyboard 7
Working with Disks 9
Loading Logo 10
Degrees 13
Using Logo 15
Play Time 24

Chapter 2
Colors and Other Things 27
Colors 27
Writing Procedures 29
The Repeat Command 33
A Magic Number 36
Saving Your Work 41

▲ Chapter 3 ▲ Circles, Variables, and Pictures 47
Circles 47
Variables 51
Saving and Reading Pictures 55

▲ Chapter 4 ▲ Over and Over 59
Loops 60
Conditions 61

▲ Chapter 5 ▲ Logo Utility Disk 67
Words 67
Animal Game 69
Easy Logo 70
Music 71

▲ Chapter 6 ▲ Sprites 77
Sampling Sprites 77
Shape Files 80
Stretching Sprites 81
The Sprite Editor 84

▲ Chapter 7 ▲ Fun Time 89
Easy Procedures 89
Stars 91
Squares 92
Triangles 94
More Squares 96
Truck 97
Houses 98

▲ Appendix A ▲ List of Commands 101

▲ Appendix B ▲ Other Reading on Logo 105

▲ Appendix C ▲ Playtime Activity Solutions 107

▲ Index 115

Introduction

This book is written to help you become familiar with the Logo computer language.

Logo is not only a language that teaches the art of computer programming. Logo is also a powerful language in its own right, and at the same time, is easy and fun to learn.

In this book you will learn the correct way to turn on the computer and all of the peripherals (monitor, disk drive, etc.), how to load Logo, and all of the commands you will need to get started using Logo.

You will also learn how to write programs that will draw many pretty and colorful designs, how to write music that your turtle will play for you, and many more things, almost all of them fun.

You don't need any special knowledge to use this book. All you need to know is where the on-off switches are on the equipment you are using.

This book takes you step-by-step through the exciting world of Logo, from turning on your computer to writing your own procedures. Many illustrations are used to show you what your screen should look like along the way.

This is not meant to be a complete course in the Logo language but a clear and simple introduction.

I wrote this book on a Commodore 64 computer using the Easy Script word processor and a Commodore 1525 printer, so don't let people tell you that your computer is only a toy. It can be a valuable tool, limited only by your imagination.

I sincerely hope you enjoy this book and that it will help you enter into the wonderful world of the turtle.

Have fun.

Working with Disks

For most of you this is the first time you will be using floppy disks.

They are a fast, reliable, and easy-to-use method of storing data, but there are some precautions you should take when using them:

1. Do not bend your disks. You could damage them and lose everything that is stored on them.
2. Do not let them sit in direct sunlight. The disks could get warped (bent out of shape).
3. Do not set your disks where you can spill anything on them.
4. Do not put your disks near magnets. Magnets may erase all of the data stored on the disks.
5. Do not touch any of the shiny parts of your disks. Handle disks only by their covers.
6. Do not force your disks into the disk drive. Insert them gently and they will last much longer.
7. Keep your disks in a safe place where nothing can happen to them.
8. Make a copy of anything that is important or hard to remake.

If you follow these rules you shouldn't have any problems with your disks.

If you don't follow these rules, you are probably in for a lot of frustration.

Chapter 1



Getting Started

While you are reading this book you will find several programs to type. You should type them exactly as they are in the book, except when you see something inside the <> marks. Do not type the letters inside the <> marks, which are called brackets. Instead, press the key. For example, <f1> means press the f1 key.

System Requirements

In order to use Commodore Logo, you will need the following equipment:

- A Commodore 64 computer
- A television set (black-and-white or color) or a computer monitor (monochrome or color)
- A Commodore VIC 1541 disk drive
- Commodore Logo software
- One blank, formatted disk

If you have a printer I will show you how to print the pictures you draw onto paper, but a printer isn't necessary.

Turning on Your Computer System

Before you go any further you should know the right way to turn your computer system on. There is only one way to do it safely and it is really simple.

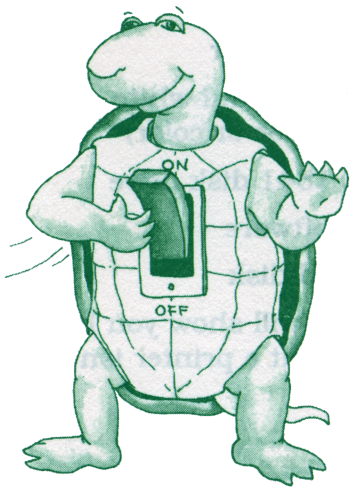
1. Turn on your printer, if you have one.
2. Make sure your disk drive is empty.
3. Turn on your disk drive.
4. Turn on your TV or monitor.
5. Turn on your computer.

If you have a disk in the drive unit when you turn it on, you could lose or scramble all of the information on the disk.

Turning off Your Computer System

When you are through using your computer you should:

1. Make sure your disk drive is empty.
2. Turn off your computer.
3. Turn off your TV or monitor.
4. Turn off your disk drive.
5. Turn off your printer.



The Commodore Keyboard ▲▲▲▲▲▲▲▲▲▲▲▲

This section will help you become familiar with the Commodore 64 keyboard. Make sure you read it carefully both before and after you load Logo. It is very important and will make your use of Logo much easier and a lot more fun.

Correcting Mistakes Before Loading Logo

The Commodore computer allows easy correction of the data that you type. In the bottom right corner of the keyboard there are two keys that enable you to move the cursor (the blinking square) around the screen. Each key has the letters CRSR identifying it as a cursor control.

Each of these keys shows two arrows on it. One key has arrows pointing up and down and the other has arrows pointing left and right.

To move the cursor down (↓), you simply press the key with the up and down arrows on it and the cursor moves down. To move the cursor up (↑), you have to hold down the <SHIFT> key and press the up and down arrow key at the same time.

To move the cursor to the right (→), you just press the key with the left and right arrow. To move the cursor to the left (←), you have to hold down the <SHIFT> key and press the left and right arrow key at the same time.

If you make a typing mistake, all you have to do is move the cursor to the right of the mistake. Then press the key marked <INST DEL> and the mistake is erased. This important key at the top right corner of the keyboard enables you to put in (INSert) or take out (DELete) information on a line.

Once you have erased the mistake, hold down the <SHIFT> key and press the <INST DEL> key. This will open up a space in the line. For each character you want to make room, you must press the <INST DEL> key once, while holding down the <SHIFT> key.

Another important key is the <RETURN> key on the right side of the keyboard, just above the cursor control keys. Nothing you type goes into the computer until you press the <RETURN> key.

With these keys you can move around the screen and make changes in the commands you have given the computer.

For example, type:

```
THE BBIG DG IS ASSLEEP
```

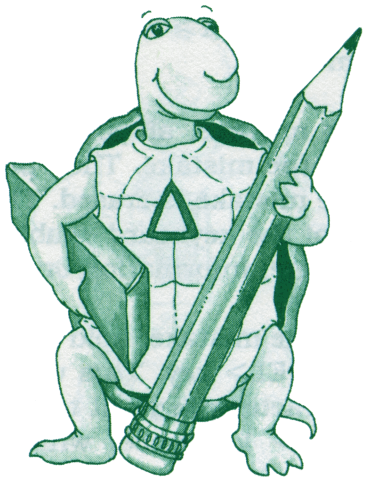
As you can see there are mistakes in this sentence.

Rather than typing the sentence all over again, move the cursor to the second B in BBIG and press the <INST DEL> key once. Did the first B disappear?

Now move the cursor to the G in DG. We meant to type DOG, so we have to put an O between the D and the G.

To open up a space in the line, hold down the <SHIFT> key and press the <INST DEL>key. Then type an O.

Now see if you can get rid of the second S in ASSLEEP.



Take some time to practice moving the cursor around the screen.

Now type something and then go back and change some of the letters you just typed. Pretty easy, isn't it?

Formatting Disks

You can format a blank disk before you load Logo, or you can wait and do it while you are using Logo. To format a blank disk before you load Logo, insert a blank disk into the disk drive and type:

```
OPEN 15,8,15
```

<RETURN>

```
PRINT#,15,"N0:disk name, id"
```

<RETURN>

Be sure to press the <RETURN> key at the end of each line. That tells the computer to enter the information you have typed on the line into its memory.

The *disk name* should be replaced with a name you have decided on for your disk, for example *Logo work disk.*

The *id* should be replaced with any two-digit number you decide to give your disk, for example *01.*

After you press the <RETURN> key the second time, your disk drive will make some noise and the red light on it will come on for a little over a minute. When the light goes out and the noise stops, the computer will prompt you with the cursor. The disk is now formatted and ready to have information saved on it.

Loading Logo

Now, using the method I just described, turn on your computer system and look at the screen. When the computer is ready the screen will display the startup message and the cursor (the blinking square). Whenever the cursor is displayed, it means that the computer is waiting for you to tell it what to do, so let's give it a job.

In the Commodore Logo package you received two disks and a book. Hold the disk marked Logo with the label up and towards your body. Insert it into the disk drive and close the door.



Now type:

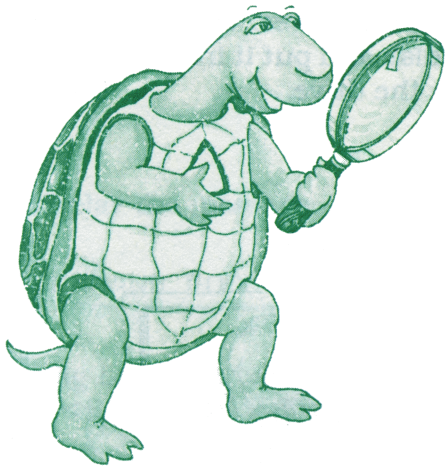
```
LOAD "LOGO",8
```

<RETURN>

After you press the <RETURN> key the red light on the disk drive will come on and there will be more noise. The screen should say:

```
SEARCHING FOR LOGO
LOADING
READY
```

If the screen shows any other message, check that you have properly inserted the right disk in the drive and closed the drive door, and that you didn't make any typing errors. Then try it again.



When the message appears on the screen as it is above, type:

```
RUN
```

<RETURN>

The screen will now say:

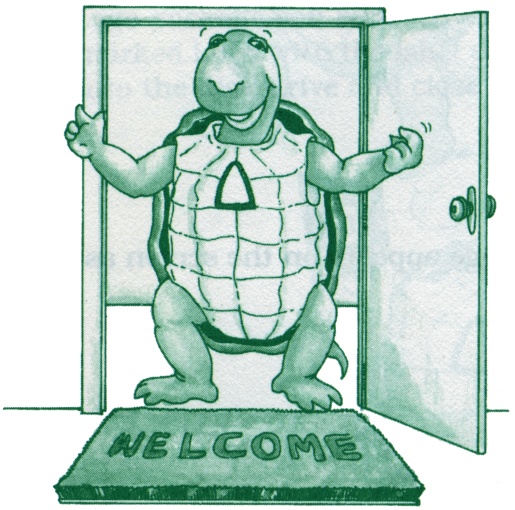
```
LOADING, PLEASE WAIT...
```

The red light on the disk drive will come on and the noise will start again. Since this will last for a minute or two, you can get a pencil and some paper in case you want to make some notes or draw some shapes before you tell the turtle to draw them.

When the light goes out and the noise stops, the screen will display the message:

```
                COMMODORE 64 LOGO
COPYRIGHT (C) 1982, 1983 TERRAPIN, INC.
COPYRIGHT (C) 1981 MIT

WELCOME TO LOGO!
?
```

If the screen shows any other message, go back to the beginning of this section and start over. When you get the right message remove the Logo disk and put it in a safe place. Then insert your formatted disk in the drive.



Correcting Mistakes After Loading Logo

It is just as easy to correct typing mistakes after you have loaded Logo as it was before. There are just a few differences.

When you make a typing mistake while using Logo you erase the mistake the same way you did before you loaded Logo. However, you don't have to open a space in the line to make room for the new characters.

For example, type:

```
FOORWRD 50
```

<RETURN>

The screen will display the message:

```
THERE IS NO COMMAND FOORWRD
```

When you are using Logo you can't move up to a line after you have pressed the <RETURN> key, but Logo will type the line over for you.

All you have to do is press the < ↑ > key, or both the <SHIFT> and the <CURSOR UP> keys, and the screen will show the line just the way you typed it.

```
FOORWRD 50
```

When the line is displayed, move the cursor over the second O in FOORWRD. Press the <INST DEL> key, and the first O is gone.

However, that isn't the only mistake in the line. We also left out the A that should go between the W and the R. To put the A in the right place, move the cursor over the R and press the A key. Logo does the rest.

We're almost ready to start using Logo. But first, in order for you to use Logo you must understand about degrees.

Degrees ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲

Degrees are easy to understand. They are just a way to measure how far away one line is from another line, using the same starting point.

Every circle, no matter how big or how small, has 360 degrees. Here is a simple example. If you have a circle 360 inches around, one inch of the circle (1/360) is equivalent to one degree.

Now if you draw lines from both ends of the inch until they meet in the center of the circle you will see that they meet at an angle. You guessed it, the angle is one degree.

Now if we measure off 90 inches of the circle and draw lines into the center of the circle the angle between the two lines is 90 degrees.

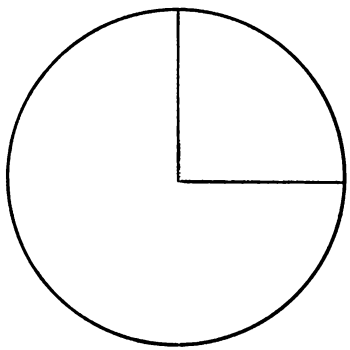


Figure 1.1 A circle with a right angle (90°) sectioned off

Here is an illustration of a circle that is a lot smaller than 360 inches. But it has 360 degrees.

An angle of 90 degrees is called a right angle.

It doesn't matter how big or how small a circle is; if you draw two lines at a 90-degree angle and draw a circle around them that uses the point where they meet for the center of the circle, the piece of the circle between the lines will be 90 degrees.

Imagine that you have a pie and you want to serve it to three friends. That means that you have to cut it into four even parts. (You want one piece for yourself.) If you cut the pie into four even pieces, each piece is 90 degrees of pie.

If you have seven friends you will need eight pieces of pie. How many degrees of pie will each person get?

(Hint: divide 360 by 8 and you will get the answer.)



Using Logo

Now type:

```
DRAW
```

Then press the <RETURN> key.

Draw is the command used to clear the screen and send the turtle back to its starting position. The screen should be divided into two parts — the top part being dark gray and the bottom part blue. If you have a black-and-white or a monochrome screen, the colors will show up as different shades of gray or different shades of the color your monitor uses.

In the center of the screen is a triangle. That's the turtle.

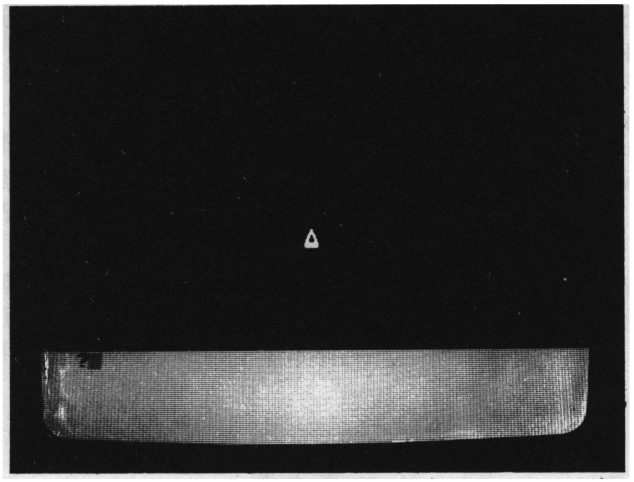


Figure 1.2 DRAW mode

You should notice that one of the sides of the turtle is thicker than the others. The angle opposite the thick side is the turtle's head.

The turtle will always move in the direction the head is pointed.

What do I mean, move?

Type:

```
FORWARD 20
```

<RETURN>

(Don't forget the space between the D and the 2.)

When you pressed <RETURN>, the turtle jumped forward 20 turtle steps and stopped.

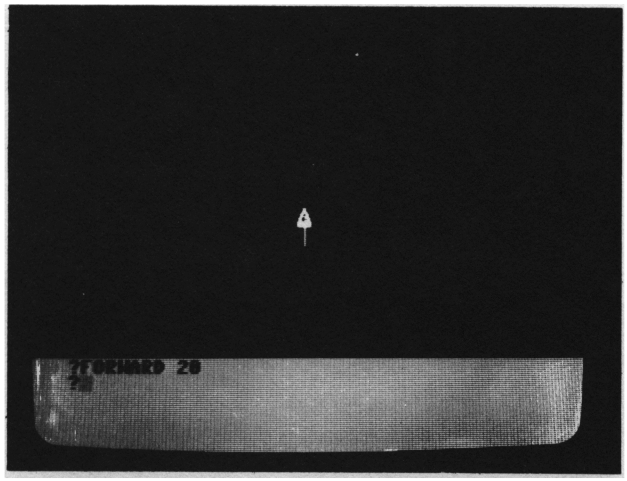


Figure 1.3 FORWARD 20

Now type:

```
RIGHT 90
```

<RETURN>

and the turtle turns 90 degrees to the right.

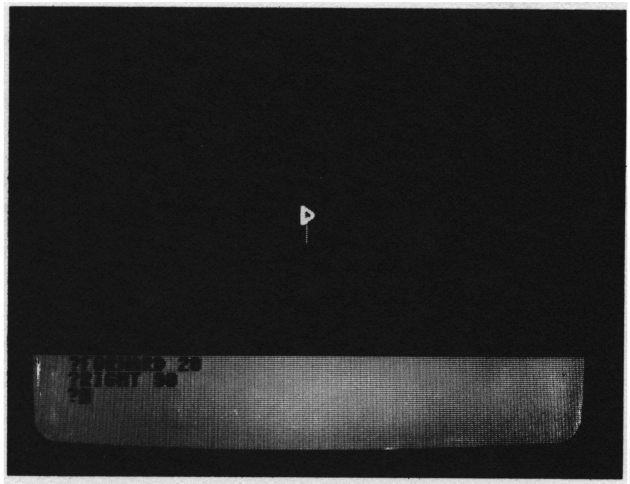


Figure 1.4 Screen after you add the RIGHT 90 command

Now type:

```
FORWARD 20
```

<RETURN>

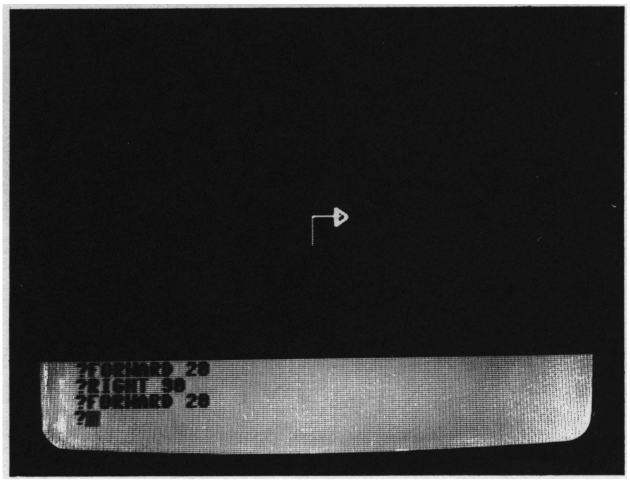


Figure 1.5 Screen after you add the FORWARD 20 command

You told the turtle to move 20 turtle steps forward and it did.

Next type:

```
LEFT 90
```

This is the last time I am going to remind you to press the <RETURN> key. You can remember to do it after completing each line.

Then type:

```
FORWARD 20
```

This time the turtle turned 90 degrees to the left and then took 20 turtle steps.

Now type:

```
FORWARD 100
```

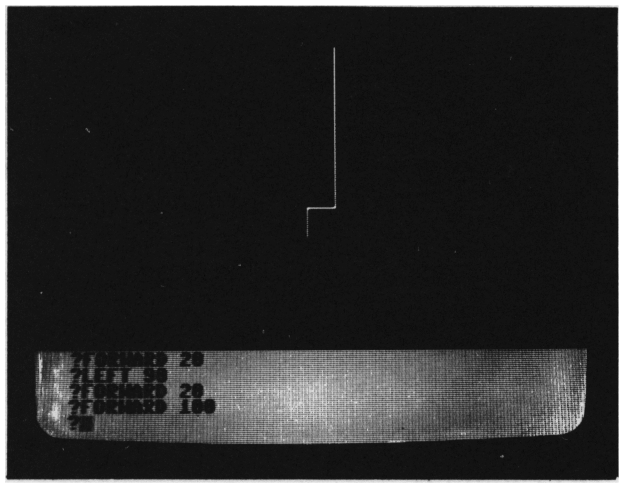


Figure 1.6 Adding the command FORWARD 100 will make your screen look like this

Where did the turtle go?

If the turtle didn't move, you probably forgot to press the <RETURN> key at the end of the line.

When you tell the turtle to go so far that it runs off the edge of the screen, it will come back on the other side. This is known as WRAPPING.

To prove that the turtle came back on the other side, type:

```
FULLSCREEN
```

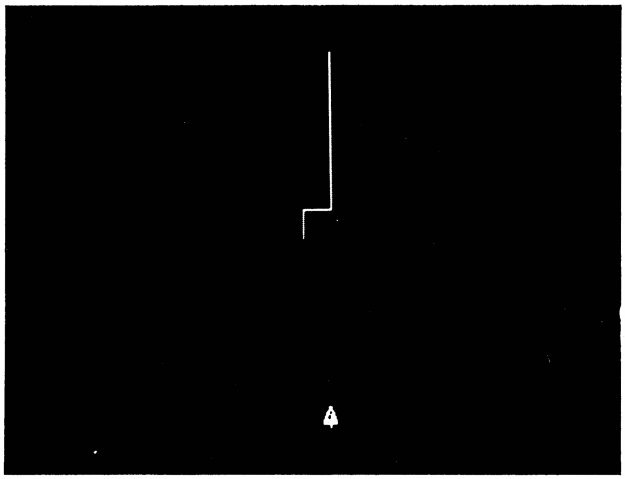


Figure 1.7 What your screen will look like if you give it the FULLSCREEN command

Now the whole screen is a drawing board for the turtle. It always was, but you could only see part of it.

Now type:

```
TEXTSCREEN
```

This time the top part (drawing screen) disappears and is replaced by the bottom half of the screen.

Now type:

```
SPLITSCREEN
```

You are back to the original setup — part of the screen for drawing and part to show what you have typed.

Now type:

```
DRAW
```

The screen clears and the turtle goes back to the center of the screen.

To keep the turtle from wrapping around the screen, type:

```
NOWRAP
```

and the turtle will stop at the edge of the screen. When the turtle stops, so does any procedure that is running. I'll tell you about procedures a little later.

First type:

```
WRAP
```

Now you have everything back to normal.

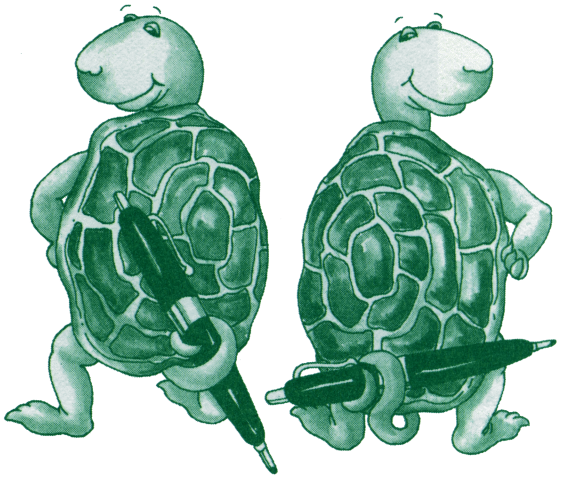
Type:

```
DRAW

PENUP

FORWARD 50
```

This time when the turtle moved forward it didn't draw a line. Do you know why?

Imagine that the turtle has a pen tied to its tail.

If you leave the turtle alone the pen touches the screen. Whenever the turtle moves the pen makes a mark on the screen.



When you give the PENUP command, the turtle raises the pen and doesn't leave a trail.

Now type:

```
PENDOWN

FORWARD 25
```

As you can see everything is back to normal.

The turtle can only move forward and backward. To make the turtle back up, type:

```
BACK 50
```

The turtle moves backward 50 turtle steps.

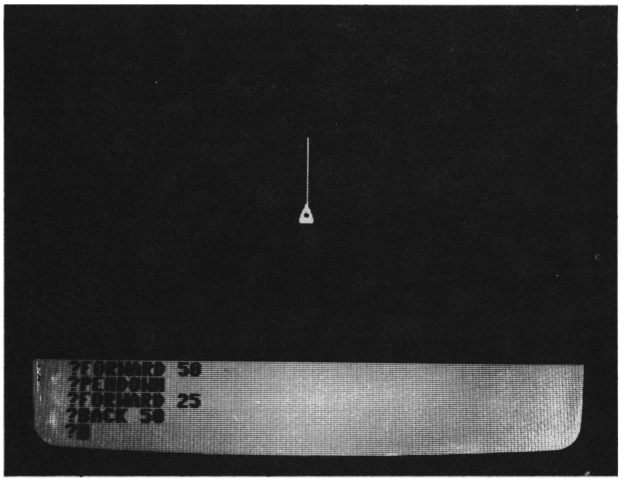


Figure 1.8 Screen after you command it BACK 50

Type:

```
HOME
```

and the turtle returns to its starting place.

Now type:

```
CLEARSCREEN
```

The screen is erased, but the turtle stays where it was.

Shortcuts

Before we go any further, I should point out that there are ways to save a lot of typing when you are telling the turtle what to do.

INSTEAD OF TYPING	TYPE
FORWARD	FD

RIGHT	RT
LEFT	LT
FULLSCREEN	<f5>
TEXTSCREEN	<f1>
SPLITSCREEN	<f3>
PENUP	PU
PENDOWN	PD
BACK	BK
CLEARSCREEN	CS
BACKGROUND	BG
PENCOLOR	PC
PENERASE	PC-1

Later, in the Easy Logo section of Chapter 5, we'll look at a program on your Logo disk that has a simplified Logo language. That language only requires one-character commands. But, for now, we'll continue looking at this more complete Logo language, along with its shortcuts for some commands. Also, a list of all Logo commands, along with what they do, is contained in Appendix A.

Play Time

Throughout this book there are sections called Play Time. These sections contain suggestions of exercises for you to do. It is my hope that you will have lots of fun doing them. If any are too hard now, don't worry. Just go on to another. Later you can go back and try again to do them. Or you can turn to Appendix C at the back of this book, where I give you hints on how to do them, and also supply you with the answers.

Now you know how to make the turtle move around the screen and draw where you want it to.

Remember, you can type DRAW to clear the screen and home the turtle any time you want.

1.1 Use the turtle to doodle on the screen.

1.2 Find out how many turtle steps it is from the bottom of the full screen to the top.

1.3 How far is it from the lower left corner to the top right corner?

1.4 Draw a square in the middle of the screen.

1.5 Draw a bigger square around the last one.

1.6 Draw a square in each corner of the screen without making any extra lines.

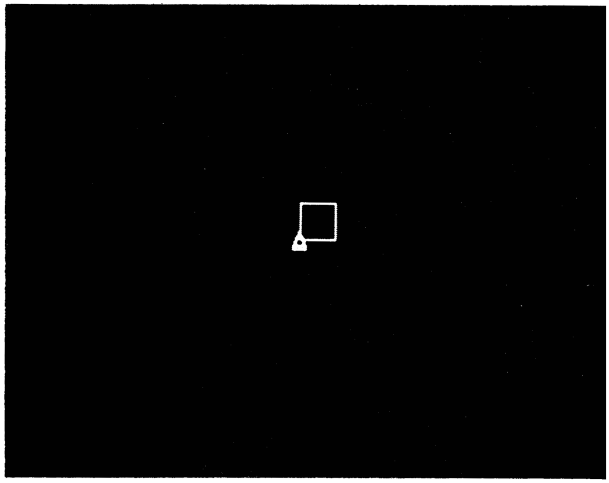


Figure 1.9 Play Time activity 1.4

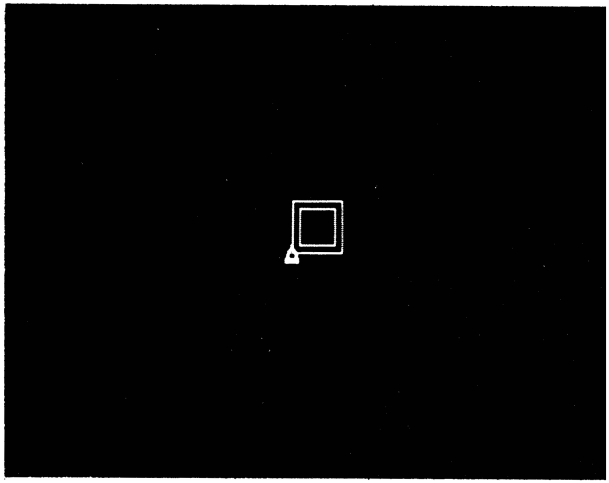


Figure 1.10 Play Time activity 1.5

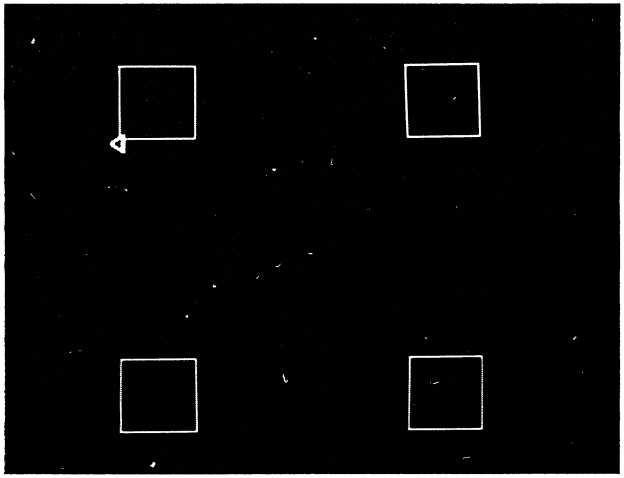


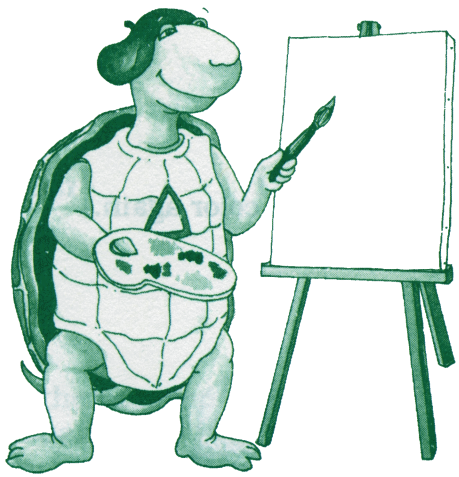
Figure 1.11 Play Time activity 1.6

Chapter 2



Colors and Other Things

Your computer has 16 colors built into it and Logo lets you make full use of all of them.



You can easily change the color of the background. To change it to blue, type:

```
BG 6
```

(BG is the shortcut for BACKGROUND. The list of Shortcuts is in Chapter 1.)

To change the color of the background to red, you type:

```
BG 2
```

To change the pencolor to blue, type:

```
PC 6
```

Change it to green by typing:

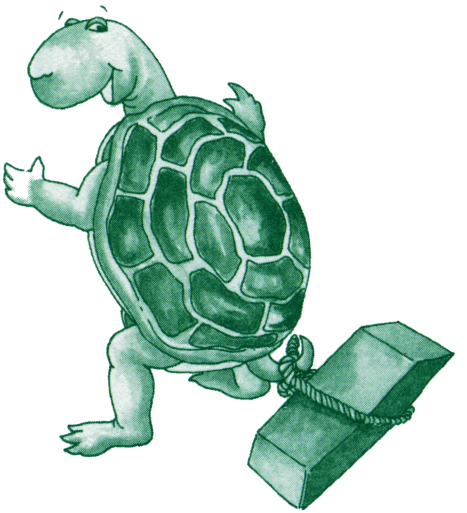
```
PC 5
```

Each computer reacts a little bit differently to colors, so you should change the pencolor and the colors of the background until you find the combination that is best for you.

Now type:

```
FD 25

PC - 1

BK 25
```

Don't forget to give the pen a color again.



Here is the complete list of colors and their numbers:

COLOR	NUMBER
Black	0
White	1
Red	2
Cyan	3
Purple	4
Green	5
Blue	6
Yellow	7
Orange	8
Brown	9
Light red	10
Gray 1	11
Gray 2	12
Light green	13
Light blue	14
Gray 3	15

Now that you know the numbers for the colors and how to use them, take some time to play with these colors on your computer. The more you play, the more you will learn.

Writing Procedures ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲

So far everything you have drawn with the turtle has disappeared as soon as it left the screen. If you drew a square and then wanted to draw another one you had to type all of the commands all over again.

In the last chapter I asked you to draw some squares, and every time you drew one, you probably did something like this:

```
DRAW

FD 25

RT 90

FD 25

RT 90

FD 25

RT 90

FD 25
```

That was okay, it worked, but every time you wanted to draw a square you had to type the whole thing over again.

To save you a lot of time and typing, I am going to show you how to define a procedure. (A procedure is a step-by-step set of directions that you give the computer.)

Don't worry, it's easy. Let's define a procedure for a square that is 25 steps long on each side.

```
TO SQUARE
```

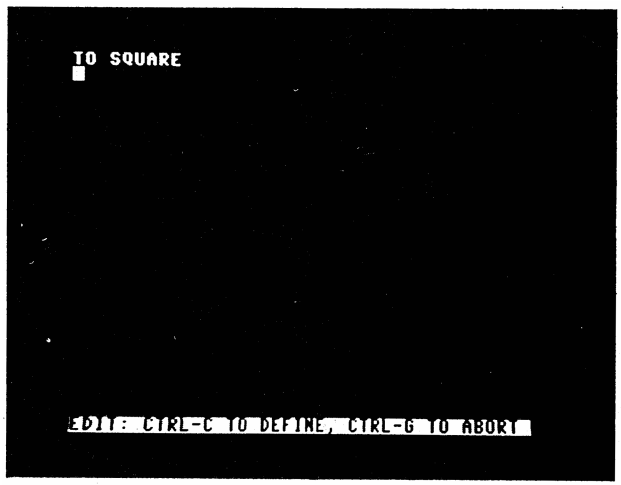


Figure 2.1 Edit screen

The screen turns gray and shows the words TO SQUARE in its top left corner. At the bottom of the screen you see:

```
EDIT:CTRL-C TO DEFINE, CTRL-G TO ABORT
```

This line lets you know that you are in the EDIT mode. There are only two ways out of the EDIT mode: <CTRL>C saves the procedure and <CTRL>G exits the EDIT mode without making any changes.

Now type:

```
FD 25

RT 90

FD 25

RT 90

FD 25

RT 90

FD 25

RT 90
```

(Of course you remembered to press <RETURN> after completing each line.)

Now, to save this procedure, press both the <CTRL> and the C keys at the same time. The screen shows the message:

```
PLEASE WAIT...
SQUARE DEFINED
?
```

SQUARE DEFINED means that your procedure named SQUARE has been stored in the computer memory. The ? means that Logo is waiting for you to give it something else to do.

Type:

```
DRAW

SQUARE
```

The turtle draws a square that is 25 steps long on each side.

Now type:

```
PU

FD 25

PD

SQUARE
```

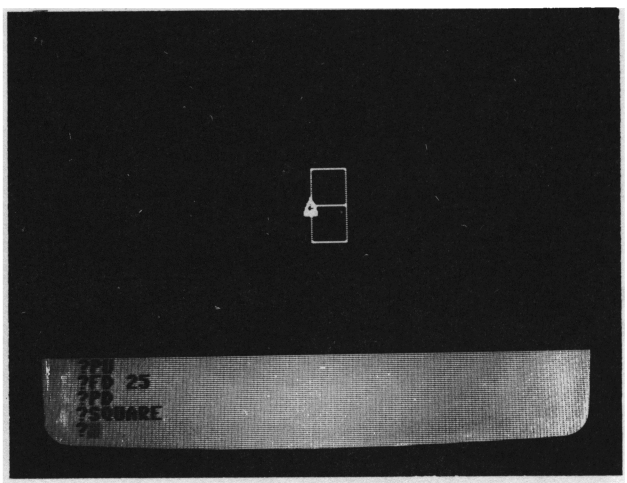


Figure 2.2 SQUARE procedure on top of another SQUARE procedure

The turtle draws another square, right above the first one.

The reason for the last turn in the procedure named SQUARE is to return the turtle to the starting position.

It is a good idea to do this whenever you can. Then it is easier for you to keep track of the turtle when you run one procedure after another.

When you define a procedure, every time you use it, you save a lot of time and typing.

Now I'll show you how to save time when you define it.

The Repeat Command▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲

Type:

```
TO SQUARE1

REPEAT 4[FD 25 RT 90]
```

Then press both the <CTRL> and C keys at the same time. When the ? prompts you for a command, type:

```
DRAW

SQUARE1
```

The turtle draws the square for you.

Using the REPEAT command allows you to repeat everything inside the square brackets [ ] as many times as you say. Make sure you use the square brackets [ ] rather than any other closure marks.

Type:

```
DRAW

REPEAT 100[FD 150 RT 15]
```

The repeat command saves you a lot of time and work, so use it whenever you can.

Another way to save time is by using the <↑>.

When you use the <↑>, Logo retypes the last command you gave the turtle.

Type:

```
DRAW

FD 25
↑
```

and you'll see what I mean.

You can put more than one command on a line, but don't forget to put a space between each different command.

Type:

```
DRAW

FD 50 RT 90 FD 50
```

Play Time

2.1. Go back to the Play Time section at the end of the last chapter. Redo the exercises numbered 4, 5, and 6, this time using your SQUARE, or your SQUARE1, routine.

2.2 Write a procedure that draws a row of squares across the screen and uses your SQUARE1 routine.

2.3 Write a procedure that draws a tower of squares using your SQUARE1 procedure.

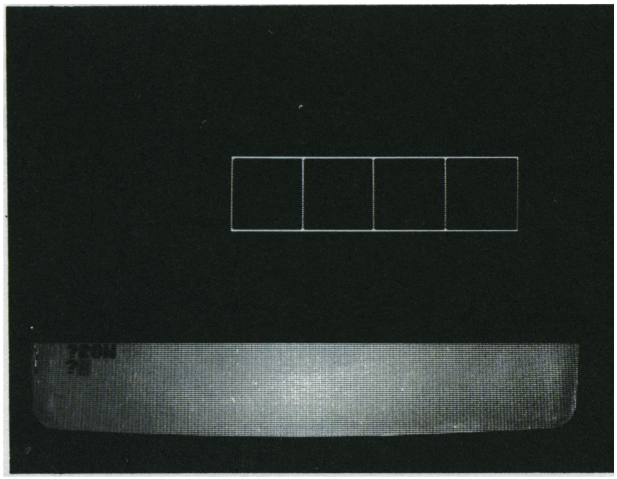


Figure 2.3 Play Time activity 2.2

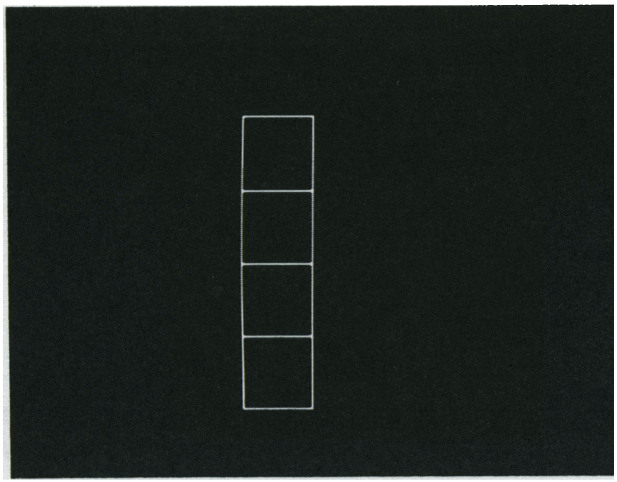


Figure 2.4 Play Time activity 2.3

A Magic Number

If you go back to the procedure named SQUARE and add the number of degrees in all of the turns you will get the answer, 360 degrees (90+90+90+90).



Since there are 360 degrees in a circle, and we want the turtle to stop where it started, we have to make it travel around 360 degrees.

Knowing this number makes it easier for us to draw different shapes. As long as the sides are all of equal length and the angles all add up to 360, the turtle will always end where it started, no matter how many sides the shape has.

To see this, type:

```
TO TRI

REPEAT 3[FD 25 RT 120]

<CTRL>C

DRAW

TRI
```

The turtle draws a triangle. (Each of the three sides is 25 turtle steps, and each of the three angles is 120 degrees.)

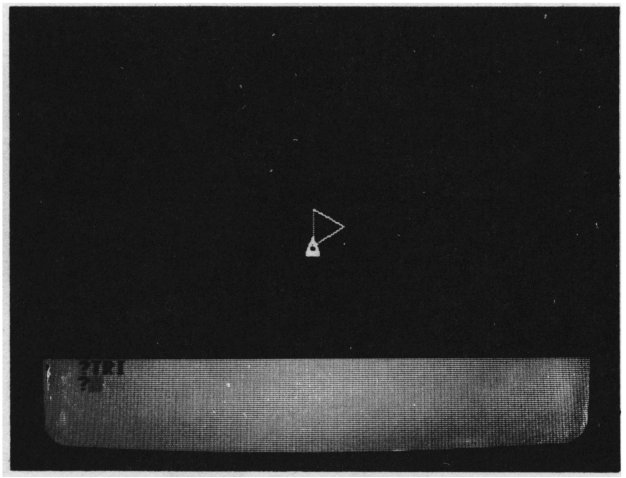


Figure 2.5 TRI procedure

If you don't want to figure out for yourself how many degrees there are in each angle of your shape, you can let the turtle do the division for you.

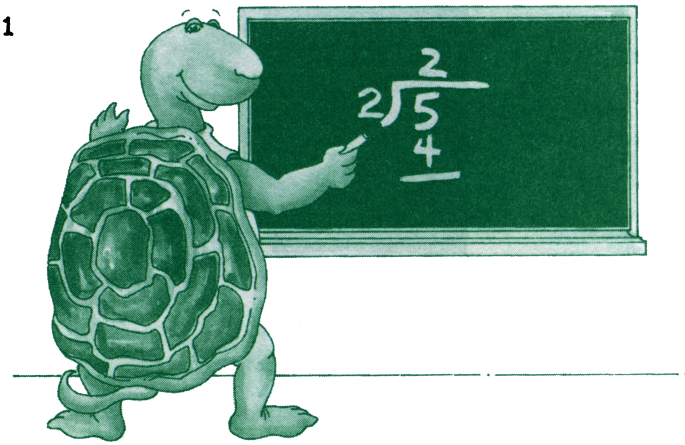
Type:

```
TO TRI1

REPEAT 3[FD 25 RT 360/3]

(the "/" means divide)

<CTRL>C

DRAW

TRI1
```

The turtle draws the same triangle as before. The only difference is that you didn't have to divide the numbers yourself.

Just for fun, type:

```
DRAW

LT 45

SQUARE1
```



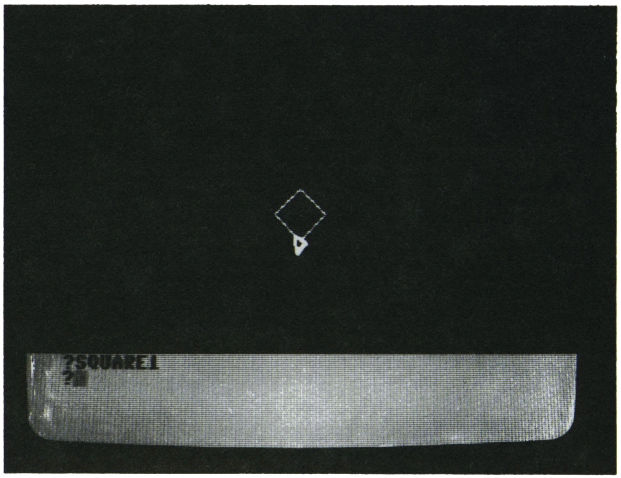


Figure 2.6
SQUARE1 procedure

Now you have a square that is resting on one of its corners instead of on one of its sides.

Once you know how to do this, there are many other things you can do. Here is one that you will like a lot. Type:

```
TO BIGSQUARE

REPEAT 4[FD 50 RT 90]

<CTRL>C

TO SPIN

REPEAT 360/10[BIGSQUARE RT 10]

<CTRL>C

DRAW

SPIN
```

That's rather pretty. Do you understand how it works?

It's really very easy. First we define a procedure called BIGSQUARE. We could have used the square procedure we already have, but I wanted a bigger square.

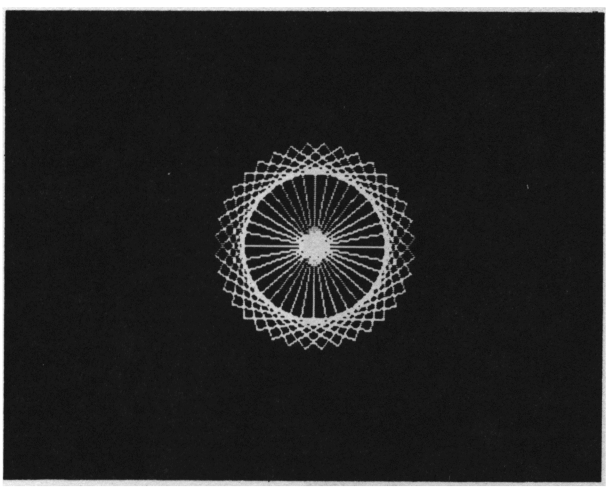


Figure 2.7 SPIN procedure

Then we defined a procedure named SPIN.

The procedure SPIN contains the repeat command, the procedure BIGSQUARE, and a turn of 10 degrees to the right.

When we give the command SPIN, we are telling the turtle to draw a square that is 50 steps on each side, then to turn clockwise 10 degrees and draw another square, and to keep doing this over and over 360/10 (36) times.

When the turtle finishes with SPIN, the ? that appears on the screen shows you that the turtle is waiting for something else to do.

Play Time

2.4 Draw a square in the middle of the screen. Now put a triangle inside the square.

2.5 Define a procedure that will draw a rectangle. (A rectangle has four sides and 90 degree angles, but usually only the opposite sides of a rectangle are the same size. When all the sides are the same size, the rectangle is a square.)

2.6 Define a procedure that will help you draw a square with an x inside it.

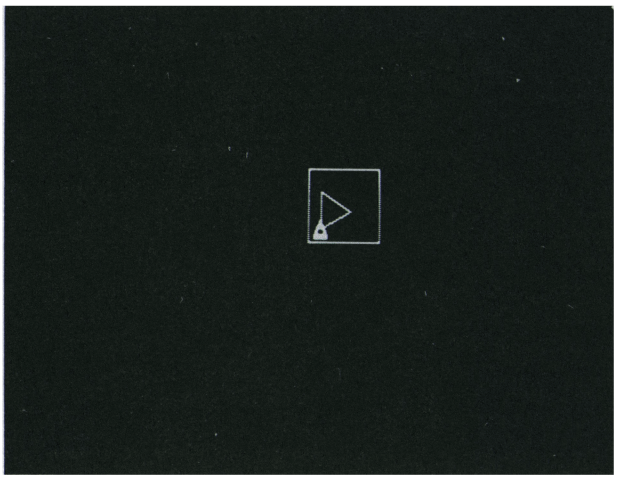


Figure 2.8 Play Time activity 2.4

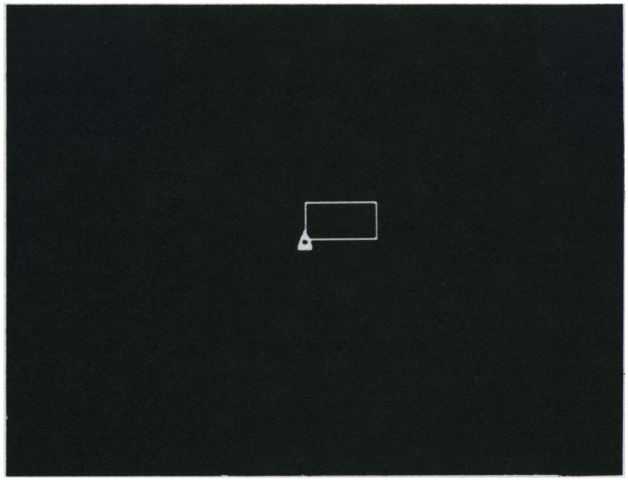


Figure 2.9 Play Time activity 2.5

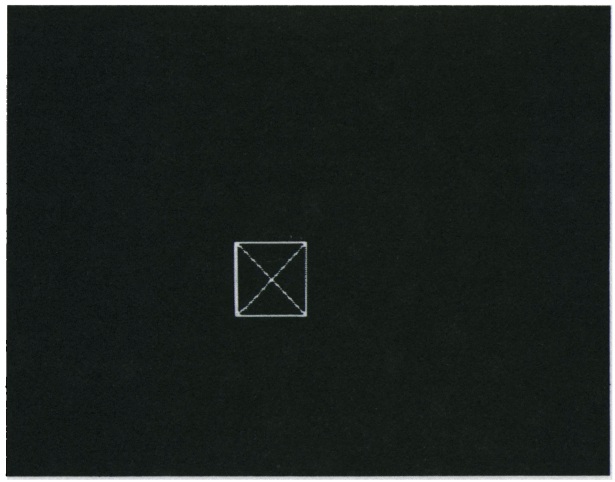


Figure 2.10 Play Time activity 2.6

Saving Your Work

After you have had some fun, taught your turtle to draw lots of pretty things and defined many procedures, what do you do with them?

If you just turn the computer off, all of your work will be lost. The next time you want to play with your turtle, you'll have to

start over from the beginning. So you don't want to do that.

The solution is to save (copy) your work onto a disk.



Take the disk that you formatted in the first chapter and insert it into the disk drive.

What's that, you didn't format your blank disk during the first chapter? Well, don't worry, you can do it now.

Insert the blank disk into the disk drive and type:

```
DOS [N0:disk name,xx]
```

The *disk name*, should be replaced with the name you have decided on for your disk, like *Logo work disk*, and the *xx* should be replaced by a two-digit number, like *01*.

When you press the <RETURN> key, your disk drive will make some noise and the red light on it will come on for a minute or two. When the light goes out and the noise stops, and the com-

puter prompts you with the ?, the disk is formatted and ready to have information saved on it.

Now that you have a formatted disk in your disk drive, you can save your procedures.

In order to save your procedures you will have to give each of them a name. You can give them any name you like, but the name should contain some hint of what the procedures are. Since all of our procedures are shapes, let's use "shapes" for our file name.

Type:

```
SAVE "SHAPES
```

Don't forget the quote mark (").

When the prompt comes back on the screen all of the procedures you had in the computer memory are now saved on the disk. If you want to test this, type:

```
CATALOG
```

and the screen will list everything you have saved on the disk. One of these things will be the file,

```
SHAPES.LOGO
```

If you haven't saved your procedures yet, save them now.

Use the command CATALOG to make sure you did it right.

Then type:

```
GOODBYE
```

Now everything that was in the computer memory is gone.

You can test this by giving the turtle a command. Type:

```
SQUARE
```

The screen will tell you:

```
THERE IS NO PROCEDURE NAMED SQUARE
```

Wait a minute! Don't get upset, you can get it all back by typing:

```
READ "SHAPES
```

As they are read into the computer memory, Logo lists all of the procedures that you saved under the name "SHAPES."

When the ? prompts you for a command, type:

```
SQUARE
```

The turtle draws the square as if nothing had happened.

Any time you are going to turn the computer off, you should save any work you want to keep.

Even if you aren't going to turn the computer off, you should save your work from time to time just in case something happens to the power supply.

What could happen?

Well, your little brother or sister could trip over the power cord, or the lights could go out, or you could forget and turn the computer off by mistake.

A lot of things can turn your turtle into a dunce. But if you have saved everything on a disk you can make him smart again.

2.7 Define a procedure that will draw a leaning stack of squares.

2.8 Define a procedure that will draw four squares so they look like a window.

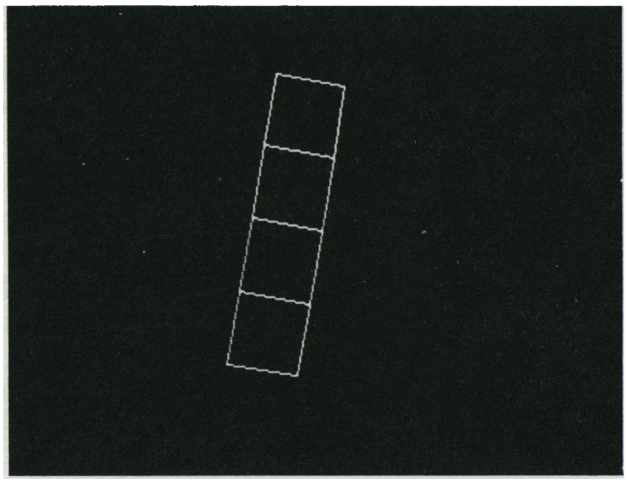


Figure 2.11 Play Time activity 2.7

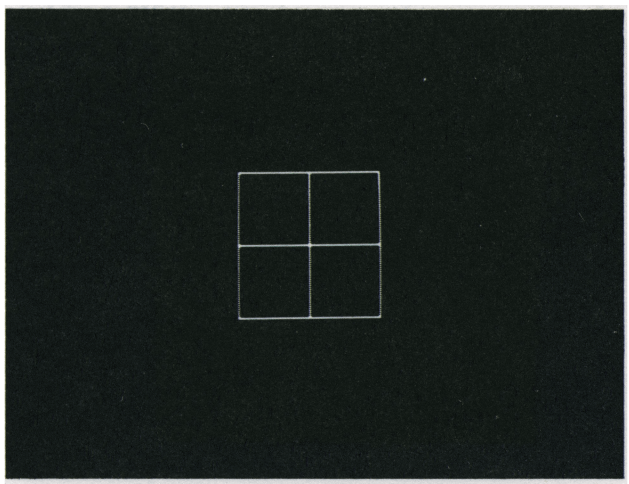


Figure 2.12 Play Time activity 2.8

Chapter 3



Circles, Variables, and Pictures

So far all of the pictures you have made your turtle draw have had straight sides and corners.

Now I am going to show you how to make your turtle draw circles.

Circles

I know that you're thinking, "If the turtle can only go in straight lines, how can it draw a circle?"

The answer is, it can't, but it can make us think it drew a circle — all it takes is a little magic (imagination).

Type:

```
TO CIRCLE

REPEAT 90[FD 1 RT 360/90]
```

and define it as usual. (You remember how: just press both the <CTRL> and the C keys at the same time.) When the ? prompt appears, type:

```
DRAW

CIRCLE
```

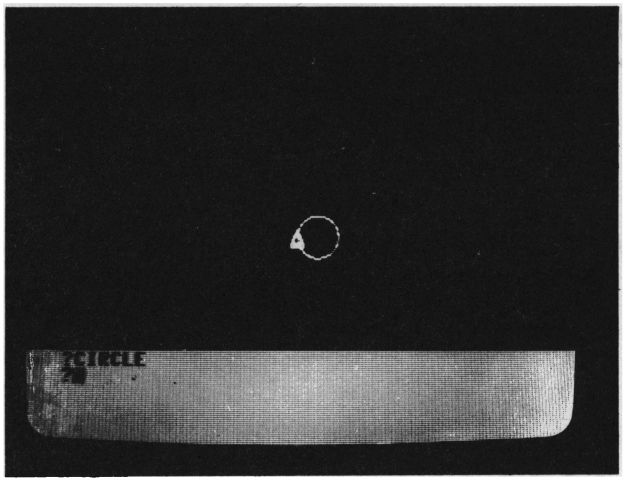


Figure 3.1 CIRCLE procedure

You and I know that the turtle didn't do anything but take one small step and make a small turn, over and over again, but let's not tell anyone else. If they want to think you taught your turtle to draw curved lines, that's okay with us.

Now type:

```
DRAW

HIDETURTLE

CIRCLE
```

Where did the turtle go?



When you tell the turtle to do something that has a lot of steps, it moves faster if you aren't watching it.

Another good use for the HIDETURTLE command is to keep the turtle from covering up part of your picture.

To see the turtle again, just type:

```
SHOWTURTLE
```

and the turtle appears as if by magic.

More Shortcuts

INSTEAD OF TYPING	TYPE
HIDETURTLE	HT
SHOWTURTLE	ST

Play Time

3.1 Change the number of turns and the size of the sides in your circle procedure and draw a lot of circles.

3.2 Draw a circle in the center of the screen. Now draw a bigger circle around the last one.

3.3 Draw a square with a circle inside.

3.4 Draw a half circle.

3.5 Draw a circle to the left.

3.6 Using the same starting point, draw three circles, each larger than the last.

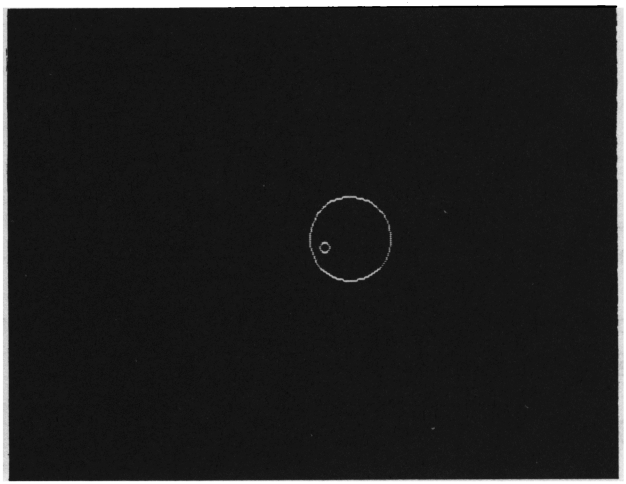


Figure 3.2 Play Time activity 3.2

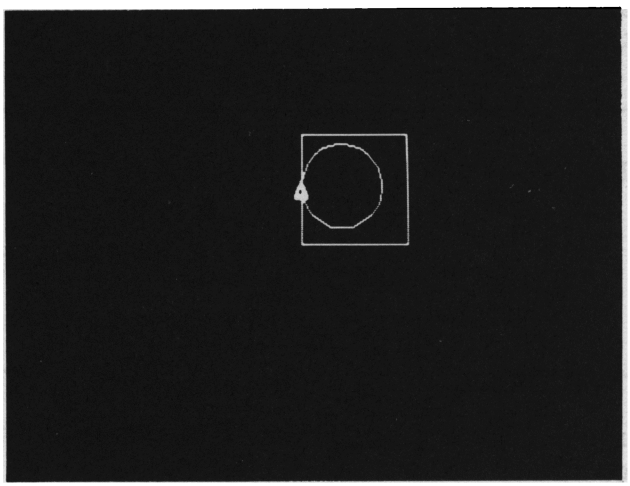


Figure 3.3 Play Time activity 3.3

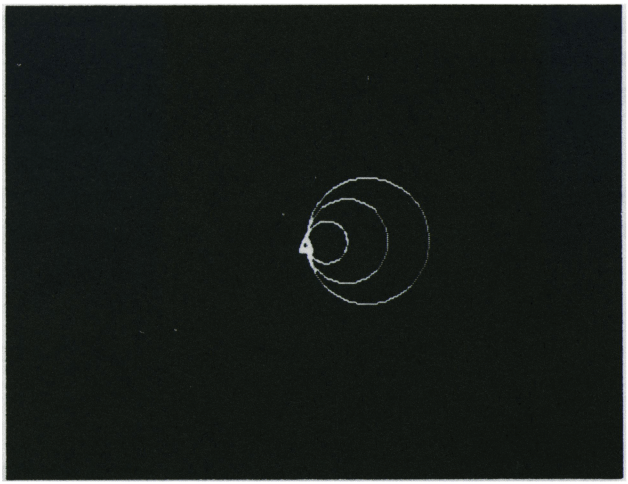


Figure 3.4 Play Time activity 3.6

Variables

Until now, every time you wanted to draw a square or any other shape bigger or smaller than the last one you have had to write a new procedure.

That takes a lot of time that you could be having fun with, so it's time for you to learn about VARIABLES.

Variables is just a big word used to describe things that change. And you can easily learn how to change them. Here's how.

Type:

```
TO SQ :LENGTH
REPEAT 4[FD :LENGTH RT 90]
```

(Don't forget to type the colon (:). It won't work without it.)

After you have defined the routine and the ? returns, type:

```
DRAW

SQ 20

SQ 30

SQ 40

SQ 50

SQ 60
```

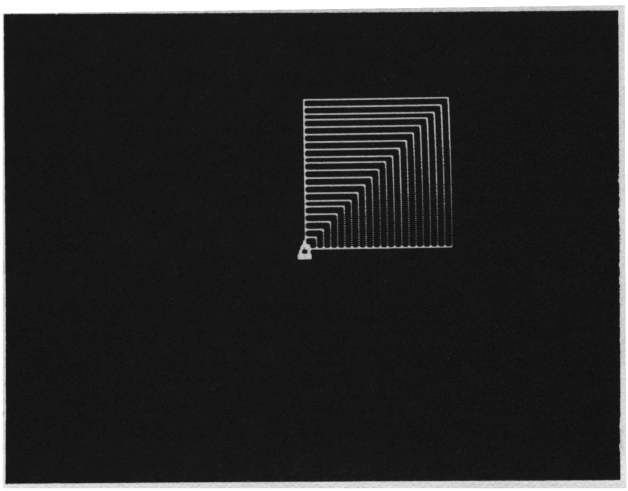


Figure 3.5 SQ procedure using the variables 20, 30, 40, 50 and 60

That's a lot easier, isn't it?

Now type:

```
SQ
```

The screen shows the message:

```
SQ NEEDS MORE INPUTS
```

Whenever you use a variable, you have to give the turtle the information it needs to do the routine. In this case, the turtle needs to know the length of each side of the square (the number of turtle steps it should move).

Sometimes you won't be able to work with Logo for a long time and you might forget what the variable is asking for. If that happens, type:

```
PO    (followed by the procedure name)
```

For example:

```
PO SQ
```

all of the steps in the procedure for drawing a square will be listed (Printed Out) on the screen.

If you forget the names of the procedures in your file, type:

```
PPOTS
```

POTS stands for Print Out Titles you will see on the screen the titles of all the procedures in your computer memory.

As I said before when we discussed procedures, remember also when using variables to give them names that have something to do with what they are being used for.

It would get confusing if we used CANDY instead of LENGTH in the SQ :LENGTH procedure.

Yes it would work, but it would be confusing.

If you don't believe it will work, type:

```
TO SQ1 :CANDY

REPEAT 4[FD :CANDY RT 90]
```

After you have defined the procedure, type:

```
DRAW

SQ1 10

SQ1 20
```

You can see that it does work.

You can use more than one variable in a procedure.

Type:

```
TO DESIGN :SIDES :LENGTH

REPEAT :SIDES[FD :LENGTH RT 360/:SIDES]
```

Define the procedure <CTRL>C, then type:

```
DRAW

DESIGN 4 50
```

You get a square, but big deal. Okay, now type:

```
DRAW

DESIGN 8 25
```

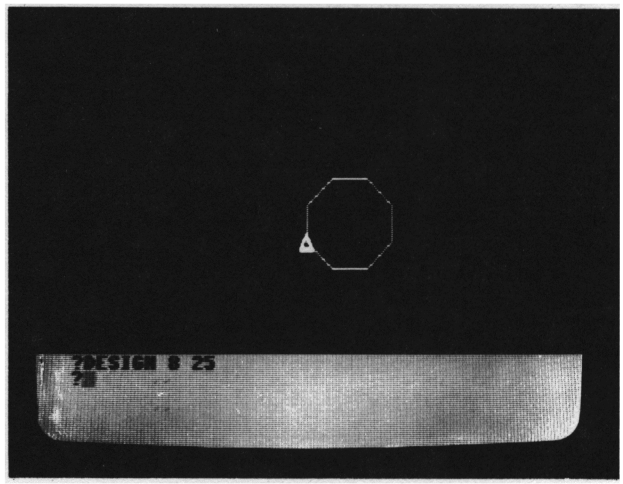


Figure 3.6 DESIGN procedure using the variables 8 and 25

Do you like that one? Try another.

Type:

```
DRAW

DESIGN 15 5
```

Play Time

I want to remind you that these Play Time exercises are included because they are fun to do. Some are very hard. If you can't do them now, just go on. Later you can come back and try again. Or you can look at the appendix in the back of this book for hints (and also answers).

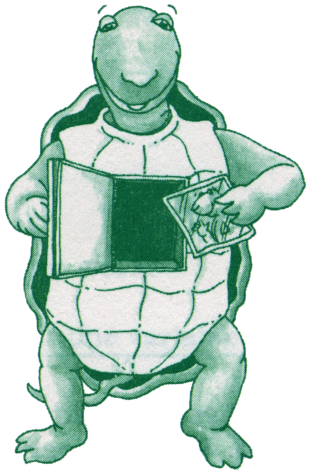
3. 7. Take some time and play with the DESIGN procedure. Try different numbers of sides and different lengths.
3. 8. Write a procedure named SPIN2 that uses a variable (:DEGREES) to set the amount you want the picture to turn, and uses the procedure DESIGN to set the shape and size of the figure.
3. 9. Write a procedure like SPIN that does the same thing with a circle.
3.10. Write a procedure that uses straight and curved lines.

Pictures

If you want to save any of the pictures that you have drawn, like one of the spin pictures, first draw the picture on the screen.

Then type:

```
SAVEPICT "SPIN2
```



It doesn't matter what name you give the picture or how you got it onto the screen. It could be a picture you made by moving the turtle around, or by running one or more procedures. When you use the SAVEPICT command, everything on the screen except the turtle is saved on the disk.

To retrieve the picture, type:

```
READPICT "SPIN2
```

The picture appears on your screen, as if by magic.

To erase the picture from the disk, type:

```
ERASEPICT "SPIN2
```

But be careful: once you press <RETURN> the picture is gone forever.

Printing Pictures

In order to print pictures you need a printer that will work with and is hooked up properly to the Commodore 64 computer.

If you have a printer, the first step is to copy the picture onto the disk by using the SAVEPICT command.

When you have saved the picture, make sure you have also saved all the procedures you want to keep. Then remove your work disk from the disk drive and put it in a safe place.

Turn off your computer, wait about 30 seconds, and then turn the computer back on.

In the package that Logo came in, there is a disk marked LOGO UTILITY DISK. Insert it properly into the disk drive and close the door.

Then type:

```
LOAD "PRINTPICT",8

RUN
```

When the red light on the disk drive goes out, remove the utility disk from the drive and replace it with the work disk.

Then type the name of the picture you want to print. For example, if you have a picture saved under the name SPIN2 and you want to print a picture of it on paper, you should type:

```
SPIN2
```

After a few seconds the printer will start to work and in just a few minutes you will have a copy of your picture to show your friends.

One thing to remember when you start saving pictures is that they take up a lot of space on the disk, about 10 times as much as the procedure that draws them, so don't save a lot of pictures unless they are really nice.

Remember, if you have written and saved a procedure, the turtle can draw the picture for you again and again.

Chapter 4



Over and Over

So far your turtle has always done exactly what you wanted, and stopped when it was finished.

Turtles are very obedient and will always do what you tell them to, but you might not always be telling them to do what you think you are.

Make sure you have your SHAPES file loaded into the computer and then type:

```
TO OAO

SQUARE

OAO
```

After you have defined this procedure, type:

```
DRAW

OAO
```

Now what's that crazy turtle doing? It's going over and over the square. It doesn't even slow down at the corners. Do you know why?

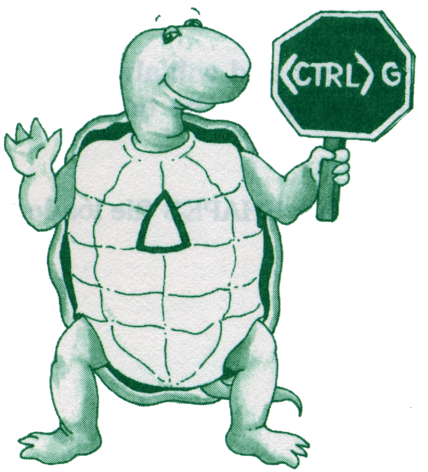
When you defined the procedure OAO (OAO is short for Over And Over), the last step was OAO.

This means that when the turtle finished the first part (SQUARE) and got to the second part (OAO) it started over again.

The process of using a procedure to refer back to itself is known as *recursion*.

To make the turtle stop running around the square, type:

<CTRL>G



No matter what the turtle is doing, you can always make it stop by typing the <CTRL> and G keys at the same time.

Loops

When the turtle was drawing the same square over and over, it was in a loop.

There are two kinds of loops you can put your turtle in. One is a closed loop and the other is an open loop.

The OAO procedure is an example of a closed loop.

The only way a turtle can get out of a closed loop is for you to tell it to stop.

An open loop has an escape hatch built into it so that when a certain condition is true, the turtle can get out of the loop all by itself.

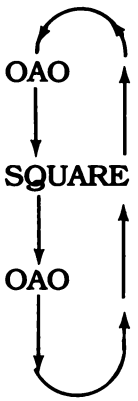


Figure 4.1 Closed loop

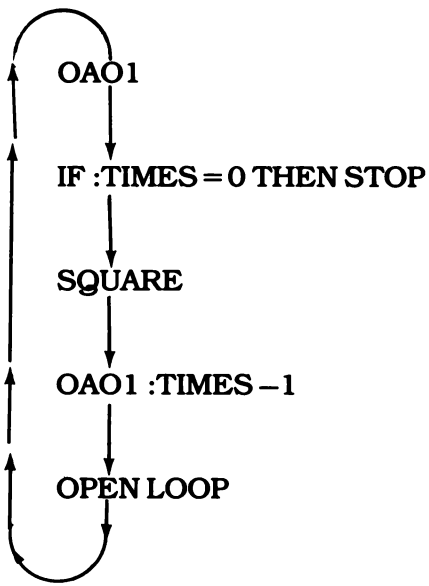


Figure 4.2 Open loop

Conditions

If you use the <CTRL>G to stop the turtle, you have to watch to make sure the turtle stops when you want it to.

There is a better way to stop your turtle.

Type:

```
TO OAO1 :TIMES

IF :TIMES = 0 THEN STOP

SQUARE

RT 15

OAO :TIMES - 1

<CTRL>C

DRAW

OAO1 5
```

In this example, you have given the turtle a condition to check for. The condition is, does :TIMES equal 0? We use the expression :TIMES — typed just like that — to tell the turtle we want it to count how many times a loop is done. We use numbers with the expression, and we tell the turtle that we want it to do something else once a loop is done that number of times.

When the turtle starts :TIMES equals five. Since five does not equal 0, the turtle draws a square.

When the turtle finishes the square it goes on to the next step, and turns right 15 degrees.

The next step tells the turtle to subtract 1 from :TIMES and go back to OAO1. 5−1=4, and 4 does not equal 0, so the turtle draws another square and turns another 15 degrees to the right.

The turtle keeps going around the loop until TIMES=0.

When :TIMES equals 0 the condition is true and the turtle stops.

Run the OAO1 procedure a few more times with different numbers to see how the turtle acts.

If, while you were playing with the OAO1 procedure, you used a negative number like −3 for :TIMES, or subtracted a number other than 1 from :TIMES, good for you. If you did not do that, try it now. (Don't forget that <CTRL>G will always stop the turtle in its tracks.)

If you tell the turtle to check for a certain condition and then stop, and that condition never happens, the turtle will keep on going until you stop it. So you have to be careful when you choose what to tell the turtle to look for.

Since we now want the turtle to check if :TIMES is less than one, let's change our procedure.

Type:

```
TO OAO1 :TIMES
```

The old procedure is shown on the screen.

Move the cursor down to the space just to the right of the 0 in the line that reads:

```
IF :TIMES = 0 THEN STOP
```

Press the <DEL> key three times. This erases the 0, the space, and the =.

Now type:

```
< 1
```

The line should look like this:

```
IF :TIMES < 1 THEN STOP
```

The < means less than.

Move the cursor to the last line of the procedure. Then define the procedure in the usual way.

Now type:

```
DRAW

OA01 5
```

This procedure seems to work just the same as the other.

Type:

```
DRAW

OA01 -1
```

The turtle doesn't draw even one square. Well, −1 is less than 1, isn't it?

Arithmetic Signs

The turtle can do all of the standard arithmetic functions. Here is a list.

SIGN	FUNCTION	EXAMPLE
+	Add	6+3=9
−	Subtract	6 −3=3
*	Multiply	6*3=18
/	Divide	6/3=2

Logo uses the * as its multiplication sign so the turtle won't get confused by the more usual sign ×.

The turtle can check for several different conditions, many more than we have used so far. Here is a list of their signs:

Condition Signs

SIGN	MEANS	EXAMPLE
=	Equals	:TIMES = 5
<	Less than	:TIMES < 1
>	Greater than	:TIMES > 1
<>	Not equal	:TIMES <> – 1
<=	Less than or equal to	:TIMES <= 1
>=	Greater than or equal to	:TIMES >= 4

For even more ways to use conditions and recursion, see the book that came with Logo.

Play Time

4.1. Spend some time playing with the procedure OAO1. Change the inputs and conditions you have the turtle check for. Observe how they affect the way the procedure works.

4.2. Use recursion in a procedure that draws a window.

4.3. Write a procedure that draws larger and larger squares. You should be able to set the size of the first square and determine how much larger each square will be.

This has been and probably always will be the hardest section in this book. If you understand this section and did all the exercises in Play Time, you are well on your way to being a turtle master.

Chapter 5



Logo Utility Disk

In the package that your Logo program came in, you will find a disk called Logo Utility Disk.

This chapter goes over some of the programs on the utility disk and gives you some helpful hints on their use.

Words

One of the important features of Commodore Logo is its word handling capability.

You can write your own Logo language.

You can even write a program that will change whatever you type into Pig Latin.

That's right, your turtle can speak Pig Latin.

Save any new procedures that are in your computer memory onto your work disk.

Remove your work disk from the disk drive and put it someplace safe.



Properly insert the Logo Utility Disk into the disk drive. To erase everything in the computer memory and make room for the new program, type:

```
GOODBYE
```

Now type:

```
READ "PIG
```

When the ? appears on screen, type:

```
PIG
```

The computer will ask you for a sentence.

When you have typed your sentence, without a period at the end, and pressed <RETURN>, the turtle rewrites your sentence in Pig Latin.

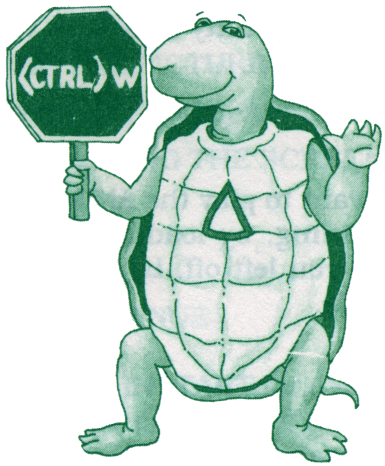
If you want to see how "PIG" works, type:

```
PO ALL
```

The entire program will be listed on your screen.

Sometimes, when Logo is listing a long procedure on the screen, the lines of the procedure run off the top of the screen before you can read them.

You can stop a long procedure from running off the top of the screen (scrolling) by typing <CTRL>W.



Just press any other key and the list will start scrolling again.

Animal Game

Also on the utility disk is the game called ANIMAL. You can easily learn to play this game and find out more about your computer's word handling and recursive techniques. With the utility disk in your disk drive, type:

```
GOODBYE

READ "ANIMAL
```

The computer will ask you to think of an animal.

When you have, the computer will ask you questions and try to guess the animal you are thinking of.

If the computer guesses wrong, it will then ask how your animal is different from the animal it guessed.

Every time the computer guesses wrong, it adds that animal to its memory.

If you want to save all the things you taught your turtle about animals, first replace the utility disk with your work disk, then type:

```
SAVE "ANIMAL
```

The next time you want to play the ANIMAL game, you can start either from the begining, by loading ANIMAL from the utility disk, or from where you left off, by loading ANIMAL from your work disk. You decide.

Easy Logo ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲

The utility disk also has an example of a simplified Logo language that only requires one-character commands.

If there are any young (nonreading) children in your family, this is the easiest program for you to use to get them started using Logo and the computer. Type:

<CTRL>G

```
GOODBYE

READ "INSTANT
```

The program starts by itself, and the screen shows:

COMMAND	ACTION
F	MOVES THE TURTLE FORWARD
R	TURNS IT RIGHT
L	TURNS IT LEFT
D	DRAW (CLEARS THE SCREEN AND HOMES THE TURTLE)
U	UNDO (MAKES THE SCREEN LIKE IT WAS BEFORE THE LAST COMMAND)
N	NAMES THE PICTURE
P	SHOWS A PICTURE, ASKS FOR ITS NAME
C	FOR CHANGING THE SCREEN COLOR
?	GIVES HELP

PRESS ANY KEY TO CONTINUE

After you have played with INSTANT for a while, you might want to modify it. If you do, just list it using the PO command, study how it is written, then make your changes.

For more information on these programs, look in the manual that came with Logo.

Music

Commodore Logo makes it easy to teach your turtle to play music by including a music editor program on the utility disk.

If you don't already have Logo loaded into your computer, do it now.

If you do have Logo loaded and have been using it, the first thing to do is to save any procedures you want to keep onto your work disk.

Then type:

<CTRL>G

```
GOODBYE
```

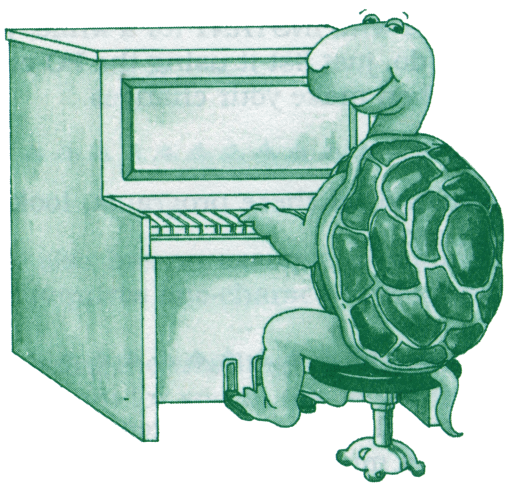
This will clear the computer memory and make room for the music program.

Remove your work disk and put it in a safe place.

Insert the Logo Utility Disk and type:

```
READ "MUSIC

READ "TWINKLE

STAR
```

You probably figured out right away what tune the turtle was going to play.



To see how the turtle knows which notes to play and how long to play them, type:

```
PO STAR
```

The screen shows:

```
STAR
STAR1
STAR2
STAR3
STAR3
STAR1
STAR2
END
```

Now type:

```
PO STAR1
```

The screen displays:

```
TO STAR1
 PLAY [0 0 7 7 9 9 7] [8 8 8 8 8 8 24]
END
```

The command PLAY is a procedure on the utility disk that lets you write music for the turtle to play.

The numbers in the first series represent the musical notes, and the numbers in the second series show the length of time the turtle waits between notes.

To see how this works, type:

```
TO STAR1
```

The screen shows the same display as before. But now change every other number in the second series to 20 and, after defining the procedure, type:

```
STAR
```

This time the tune sounds completely different.

Again type:

```
TO STAR1
```

Change the numbers in the second series back to the way they were when you first saw them. After you have done that, change the first two 0s in the first series of numbers to 5s, define the procedure, and then type:

```
STAR
```

This time the first two notes are higher in pitch and don't fit the song.

I have included a few tunes that you should be able to recognize. However, there are some slight mistakes in them.

When you type them in and run them, you will probably want to change them slightly to see if you can make them sound better. Go ahead, it's good practice.

Type:

```
GOODBYE

READ "MUSIC

TO NA

PLAY [4 2 0 2 4 7 10 9 8 2 3 4] [12 3 24 24 24
  48 12 3 24 24 24 48]
```

Define the procedure and type:

```
NA
```

One way to get the turtle to play music is to put all of the numbers one right after another. A better way is to break the numbers up into several pieces.

Type:

```
TO CLEM

PLAY [3 3 3 0 5 5 5 3] [5 5 10 10 5 5 10 10]

PLAY [3 5 7 7 6 5 4] [5 5 10 10 5 5 20]
```

Define the procedure, and type:

```
CLEM
```

In this example, we have divided the tune up into two parts, but that's all we have done. The two parts are still connected as part of the CLEM routine.

The third and best way is to give the different parts of the tune separate names.

That way they can be used as subprocedures in a main procedure. For example, type:

```
TO OS

OS1

OS2
```

Define the procedure, and type:

```
TO OS1
```

Define the procedure, and type:

```
TO OS2
PLAY [1 1 3 5 8 6 6 4 5 6 5] [6 6 6 6 18 30 6
 6 6 6 48]
```

After you have defined the procedure, type:

```
 OS
```

See if you can tell which song it is.

The main reason for defining the different parts of a tune as separate procedures is so you can use them in a different order.

Type:

```
READ "TWINKLE

PO STAR
```

You can see that the different procedures are used more than once. You save a lot of typing and time when you separate the parts of a tune so you can repeat any part without retyping it.

For more information on how Logo plays music, read the section on music in the book that came with your Logo package.

Chapter 6



Sprites

Your Commodore 64 gives you the ability to design up to eight programmable figures, called SPRITES.

Sampling Sprites

Logo takes full advantage of this feature.

The sprites are numbered from 0 through 7 and you can use them in programs.

Insert the utility disk into your disk drive, and type:

```
READ "SPRITEDEMOS

READ "DINOSAURS
```

and you will see a family of dinosaurs going for a picnic.

You can stop the program at any time by pressing the <CTRL> and G keys at the same time, or you can stop it at the end of a cycle by pressing any key.

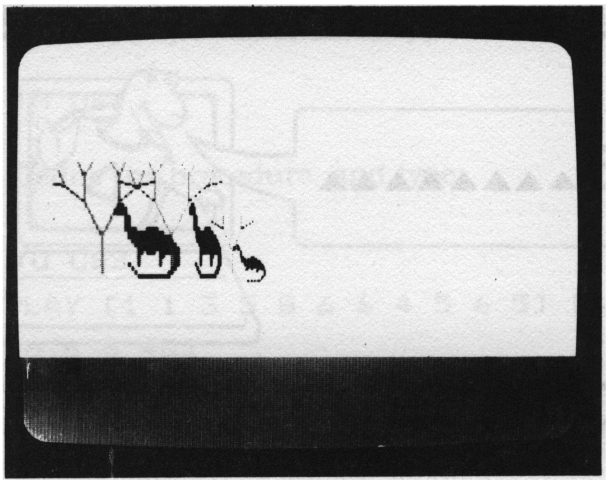


Figure 6.1 This is what the sprite program DINOSAURS looks like.

When you get tired of watching dinosaurs eating trees, stop the program and type:

```
READ "SUBMARINE
```

You can stop it in the same way you stoppd DINOSAUR.

Then type:

```
READ "RUNNER
```

You now see a sprite that looks like a woman, running across the screen.

At the end of her run she gets tired and sits down. Then all of the shapes used to make the woman appear to run are shown at the right of the screen.

By using several sprites with slightly different views of the same basic shape, you can give the illusion of movement. This is called animation.

The utility disk already has some sprite shapes on it. You can see them.

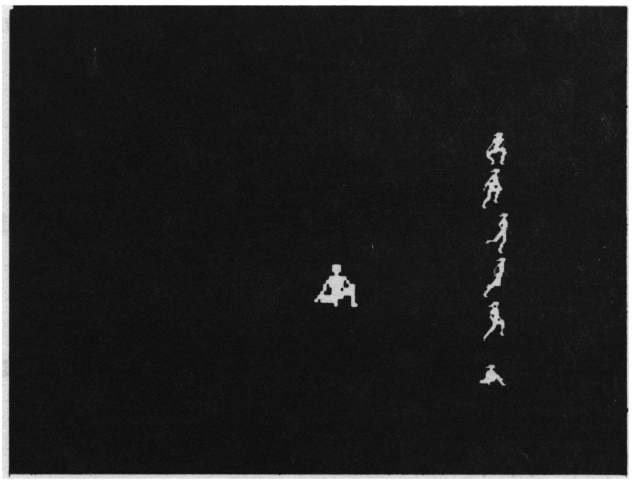


Figure 6.2 A picture of the sprite program RUNNER

First type:

```
GOODBYE
```

This makes room for the new files.

```
READ "SPRITES
```

Then type:

```
DRAW
```

Sprite number 0 is the turtle. When you type a command the turtle responds as it should.

Now type:

```
READ "ANIMALS
```

The file of sprite shapes found under the name ANIMALS is loaded into the computer memory.

In order to get an individual sprite's attention and get it to move, you have to TELL that sprite what to do.

Type:

```
TELL 1 FD 100

ST
```

You can see the dinosaur appear on the screen.

Next type:

```
TELL 2 RT 90

FD 25

LT 90

ST
```

Now you see a kangaroo on the screen.

Shape Files

There are five files of sprite shapes on the utility disk. Here are four of them, along with their corresponding shape numbers. They are listed under the names of the files they occupy on the disk. these file names are: ANIMALS.SHAPES; ASSORTED.SHAPES; SHAPES.SHAPES; and VEHICLES.SHAPES.

ANIMALS.SHAPES	ASSORTED.SHAPES
1 Dinosaur	1 Target
2 Kangaroo	2 Truck
3 Bug	3 Rocket
4 Dolphin	4 Balloon
5 Horse	5 Bow

6 Cat	6 Arrow
7 Butterfly	7 Man

SHAPES.SHAPES	VEHICLES.SHAPES
1 Frame	1 Truck
2 BBall (big ball)	2 Car
3 SBall (small ball)	3 Bicycle
4 Square	4 Submarine
5 Triangle	5 Airplane
6 Heart	6 Boat
7 Box	7 Balloon

The fifth file of shapes is the RUNNER file. It contains the shapes that you looked at when you typed runner near the start of this chapter. These shapes aren't listed here because they don't have different names like these in the other four shape files.

There are only seven shapes in each shapes file because shape 0 is always the turtle.

Play Time

Take some time and, using the TELL command, play with the different shapes.

Stretching Sprites

When you are using sprites, you can make them twice as wide by using the BIGX command and twice as high by using the BIGY command.

Type:

```
GOODBYE

READ "SPRITES

READ "ANIMALS
```

When you have read the ANIMALS file into your computer memory, type:

```
TELL 0 HT

TELL 5 ST

BIGX
```

You get a very long horse.

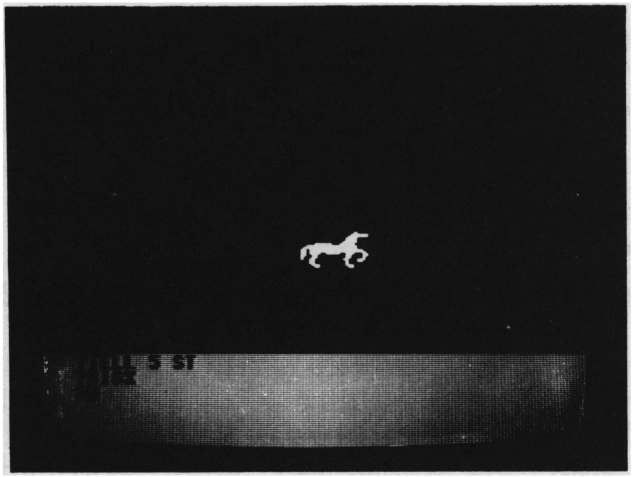


Figure 6.3 Screen of the horse after the command BIGX

Type:

```
BIGY
```

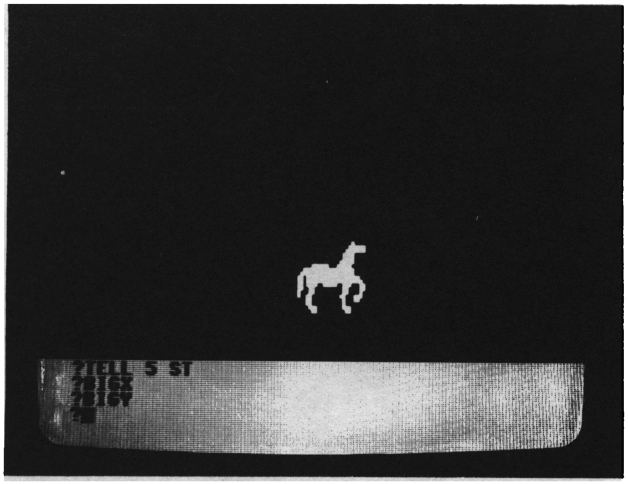


Figure 6.4 After you give the command BIGY the horse should look like this

You now get a big horse of normal proportions.

Type:

```
SMALLX
```

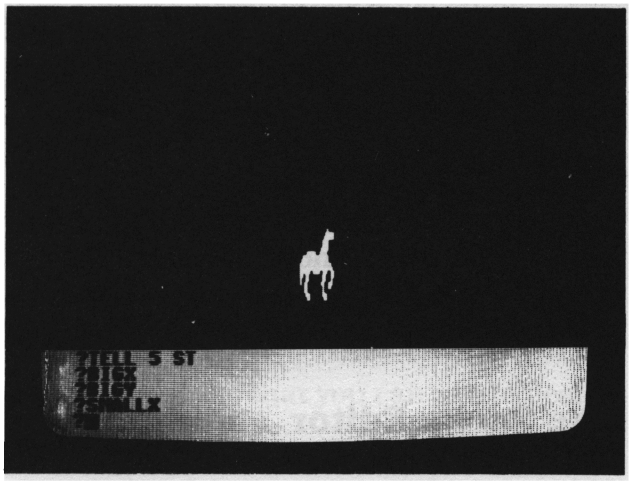


Figure 6.5 The SMALLX command will make the horse look like this

You get a very tall horse.

Now type:

```
SMALLY
```

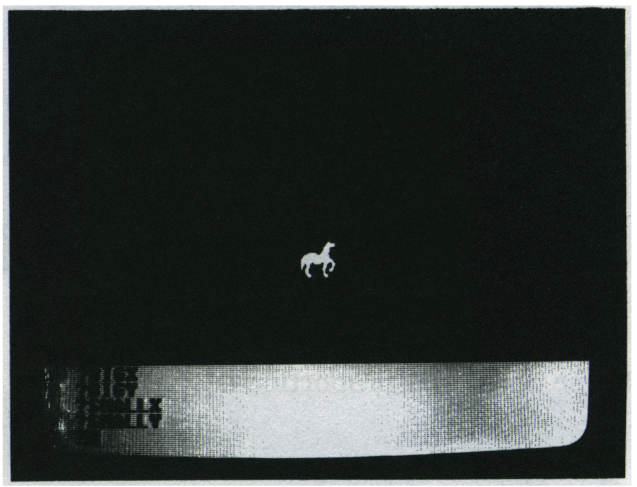


Figure 6.6 SMALLY will return the horse to its original shape

You see the horse returns to normal size.

If you don't say you want BIGX or BIGY, the sprites will be shown as small as possible; this is the DEFAULT setting.

(Default means that you haven't told the computer you wanted anything special, so you get the normal setting.)

Some other default settings you have seen are the background color in the DRAW mode and the PENCOLOR.

You can usually change default settings if you want to.

The Sprite Editor ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲

The sprite editor is a powerful utility provided by Logo that lets you design your own sprites. (An example of a sprite is any one of the shapes found in the previous section.)

With the utility disk in your disk drive, type:

```
GOODBYE

READ "ANIMALS

READ "SPRED

TELL 1

EDSH
```

The screen shows a picture of the current sprite in the center and a large box next to it with a picture of the current sprite drawn in dots and balls. The dots represent blanks and the balls represent filled-in pixels, which are tiny sections of the screen.

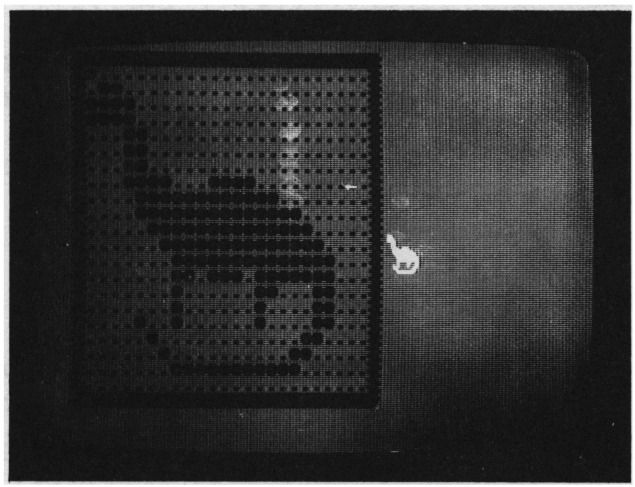


Figure 6.7 Sprite editor screen

The sprite box is 24 characters wide and 21 characters high.

These characters represent the pixels that you can use to design a sprite.

If you get very close to your screen and look carefully at it, you will see a lot of very small squares. These squares are called PIXELS. (Pixel is short for picture element.)

To move around in the sprite editor, use the <CRSR> keys, the <RETURN> key, and the <HOME> key.

To fill in a pixel, place the cursor over the dot you want to fill in, then press the <*> key.

To clear a pixel, place the cursor over the space you wish to clear, then press the <SPACE> bar.

As you make changes with the sprite editor, you can see the changes appear in the picture of the sprite on the screen.

Sometimes, when you are designing a new sprite, it will help to draw it out on a piece of graph paper.

Mark a spot 24 squares wide by 21 squares high. Then draw your sprite before you try putting it into the sprite editor.

You exit the sprite editor in the same way as the procedure editor. You press both the <CTRL> and C keys at the same time.

After you have designed a number of sprites, you can save them in a file by typing:

```
SAVESHAPES "filename
```

Use any name you like for the *FILENAME*, but be careful. If you give it the same name as a file already on the disk, the first file will be gone forever.

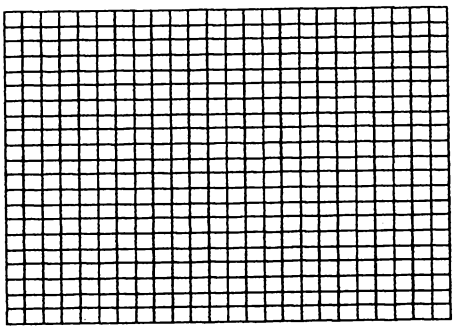
Sprites can be programmed to explode on contact with other sprites or the background, to ignore or bounce off other sprites or the background, or to pass in front of or behind other sprites.

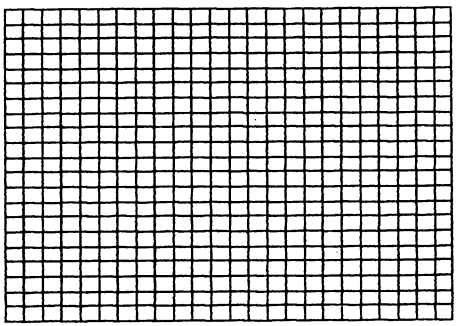
Sprites can also be taught to obey commands from a joystick.

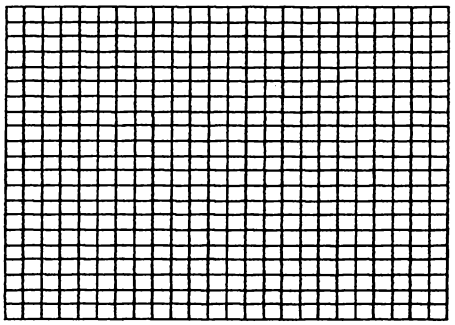
For more information on the design and use of sprites, see the chapter on sprites in the book that came with Logo.

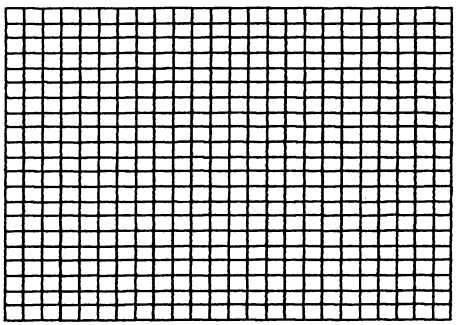
Sprite Blocks

Use these sprite blocks to help you design your sprites. You can cut this page out of the book and copy it so you will always have plenty of sprite blocks to work on.









Chapter 7



Fun Time

Easy Procedures

Now that you know how to make your turtle do what you want it to, here are some procedures you can have fun with.

After you have typed them in, defined them, and run them, see if you can figure out how they work. Then see if you can change them to make them better.

Remember that you can put more than one command on a line, and that you must press <RETURN> at the end of each line:

```
TO T

FD 50 LT 90 FD 20 RT 90 FD 5 RT 90

FD 45 RT 90 FD 5 RT 90 FD 20 LT 90

FD 50 RT 90 FD 5 RT 90

<CTRL>C
```

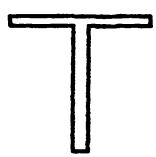


Figure 7.1 T procedure

```
TO CROSS
PU LT 90 FD 5
RT 90 FD 5 PD
REPEAT 4[FD 50 RT 90 FD 10 RT 90 FD 50 LT 90]
```

<CTRL>C

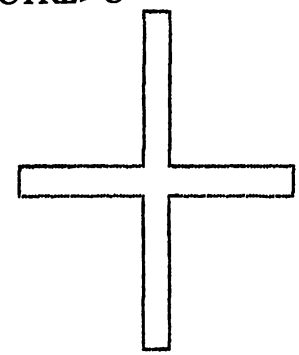


Figure 7.2 CROSS procedure

```
TO SQX1
RT 45
REPEAT 4[FD 50 RT 135 FD 70 RT 135 FD 50]
LT 45
```

<CTRL>C

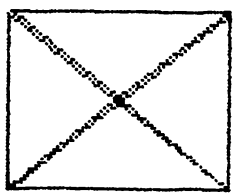


Figure 7.3 SQX1 procedure

Stars

Here are two procedures that will draw stars. Which one do you like best? Can you figure out why they look so different?

```
TO STAR

RT 33

REPEAT 5[REPEAT 3[FD 50 RT 120] FD 50 LT 360/5]
```

<CTRL>C

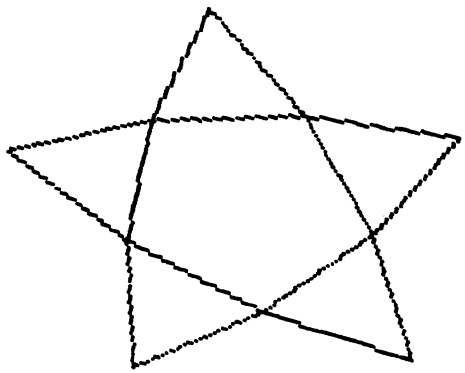


Figure 7.4 STAR procedure

```
TO STAR1

REPEAT 5[RT 17.5 FD 50 RT 145 FD 50 LT 90]
```

<CTRL>C

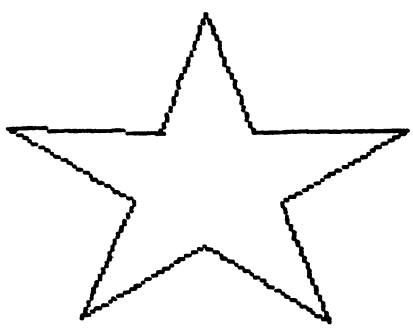


Figure 7.5 STAR1 procedure

Squares

Here are three procedures that will draw squares. They are all slightly different. Try all three and study them to see the differences.

```
TO GROW :LENGTH

IF :LENGTH > 250 THEN STOP

FD :LENGTH

RT 90

GROW :LENGTH + 3
```

<CTRL>C

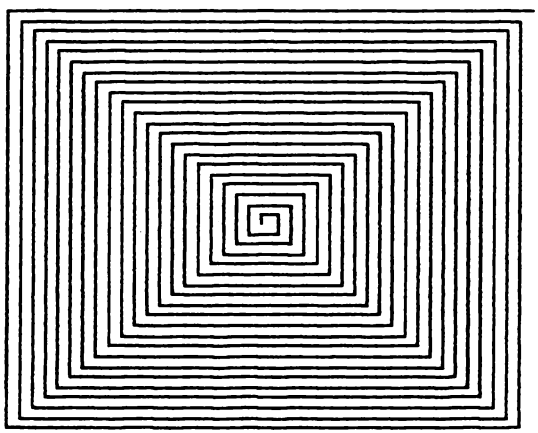


Figure 7.6 GROW procedure

```
TO GROW1 :LENGTH

IF :LENGTH < - 250 THEN STOP

FD :LENGTH

RT 90

GROW1 :LENGTH - 10
```

<CTRL>C

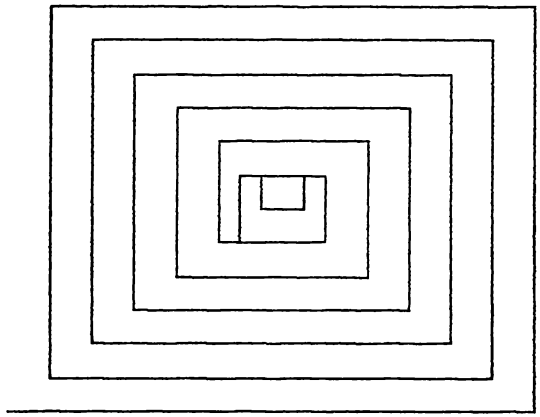


Figure 7.7 GROW1 procedure

```
TO GROW2 :LENGTH

IF :LENGTH >200 THEN STOP

FD :LENGTH

RT 85

GROW2 :LENGTH + 3
```

<CTRL>C

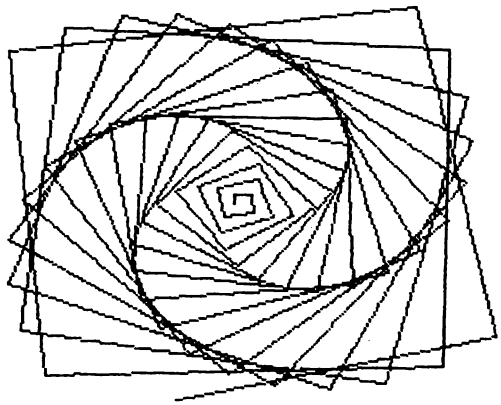


Figure 7.8 GROW2 procedure

Triangles

Here are three procedures that will draw triangle spirals. They are all different. Try them all, and study them to see how they are different.

```
TO TRIAL :LENGTH

IF :LENGTH > 250 THEN STOP

FD :LENGTH RT 120

TRIAL :LENGTH + 3

<CTRL>C
```

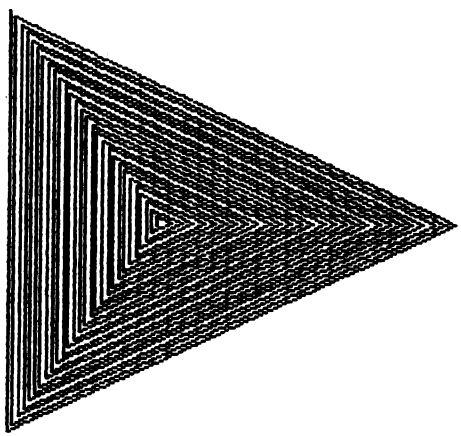


Figure 7.9 TRIAL procedure

```
TO TRIAL1 :LENGTH

IF :LENGTH < - 250 THEN STOP

FD :LENGTH RT 120

TRIAL1 :LENGTH - 5

<CTRL>C
```

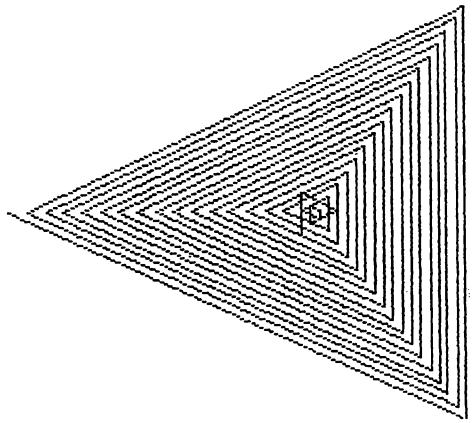


Figure 7.10 TRIAL1 procedure

```
TO TRIAL2 :LENGTH

IF :LENGTH > 250 THEN STOP

FD :LENGTH RT 117

TRIAL2 :LENGTH + 3
```

<CTRL>C

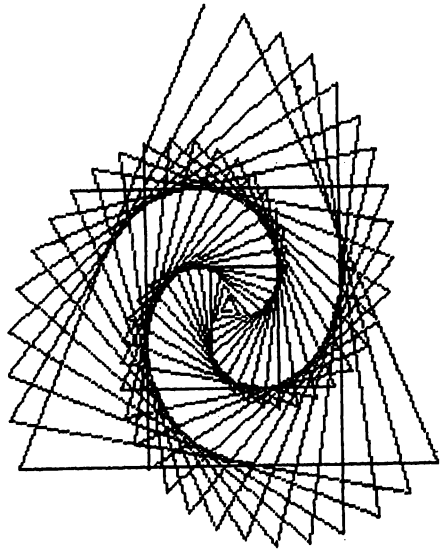


Figure 7.11 TRIAL2 procedure

More Squares

Here are two procedures that draw bigger and bigger squares:

```
TO BIGGER :LENGTH
IF :LENGTH > 130 THEN STOP
REPEAT 4[FD :LENGTH RT 90]
BIGGER :LENGTH + 10
```

<CTRL>C

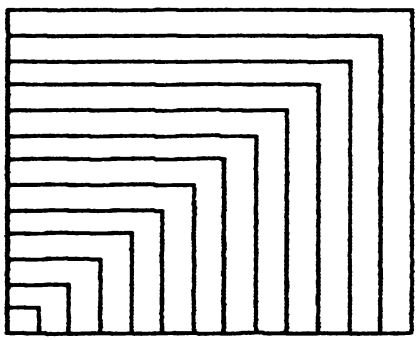


Figure 7.12 BIGGER procedure

```
TO 4BIGGER :LENGTH
REPEAT 4[BIGGER :LENGTH RT 90]
```

<CTRL>C

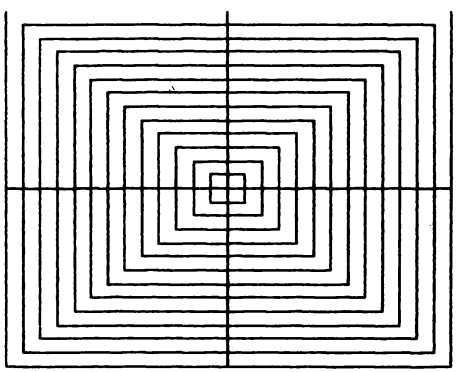


Figure 7.13 4BIGGER procedure

Truck ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲

Here's a special shape. Remember to press <RETURN> at the ends of the lines:

```
TO PUTRUCK

HT PU LT 90 FD 150 RT 90 PD FD 50 RT 90 FD 50
   LT 60 FD 50 RT 60 FD 50 RT 90

FD 50 LT 90 FD 150 RT 90 FD 45 RT 90 FD 45
   RT 33 REPEAT 66[FD 1 LT 1]

RT 33 FD 90 RT 33 REPEAT 66[FD 1 LT 1] RT 33
   FD 15 PU BK 45 RT 90 FD 7 LT 90

PD REPEAT 360/3[FD 1 LT 3] PU BK 150 PD REPEAT
   360/3[FD 1 LT 3] PU LT 90

FD 12.5 RT 90 PD REPEAT 360/10[FD 1 LT 10]
   PU FD 150

PD REPEAT 360/10[FD 1 LT 10] PU RT 90 FD 62.5
   RT 90 FD 15 LT 60 PD FD 40

RT 60 FD 35 RT 90 FD 85 RT 90 FD 29 RT 33
  REPEAT 30[FD 1 LT 1] RT 90 FD 40

RT 90 FD 52 PU RT 90 FD 10 RT 90 FD 5 PD FD 5
```

<CTRL>C

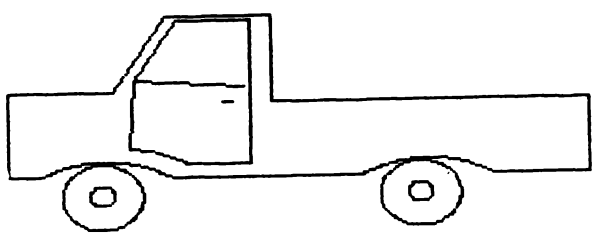


Figure 7.14 PU-TRUCK procedure

Houses ▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲

These procedures will draw a house any size you want:

```
TO PANE :LENGTH
REPEAT 4[FD :LENGTH/2 RT 90]
```

<CTRL>C

```
TO CHIMNEY
FD :LENGTH/5 RT 90 FD :LENGTH/10 RT 90 FD :LENGTH/5+2
```

<CTRL>C

```
TO ROOF
LT 40
FD ( :LENGTH + :LENGTH/10)/1.42857
LT 160 FD :LENGTH + :LENGTH/2.5
```

<CTRL>C

```
TO DOOR
FD :LENGTH/2.5 RT 90 FD :LENGTH/5 RT 90 FD :LENGTH/2.5
```

<CTRL>C

```
TO WINDOW :LENGTH
REPEAT 4[PANE :LENGTH RT 90]
```

<CTRL>C

```
TO FRAME :LENGTH
```

```
REPEAT 4[FD :LENGTH RT 90]
```

<CTRL>C

```
TO HOUSE :LENGTH
HT PD
FRAME :LENGTH
PU FD :LENGTH/3 RT 90 FD :LENGTH/5 PD
WINDOW :LENGTH/5
PU FD ( :LENGTH - :LENGTH/5) - :LENGTH/5 LT 90 PD
WINDOW :LENGTH/5
PU FD :LENGTH/2 PD
WINDOW :LENGTH/5
PU LT 90 FD ( :LENGTH - :LENGTH/5) - :LENGTH/5 PD
WINDOW :LENGTH/5
PU FD :LENGTH/5
RT 180 FD :LENGTH/2 - :LENGTH/10 RT 90
FD :LENGTH/5 * 4.15
PD RT 180
DOOR
PU RT 180 FD :LENGTH RT 90 FD :LENGTH/1.75
      :LENGTH/20
ROOF
PU LT 160 FD :LENGTH/2 RT 70 PD
CHIMNEY
```

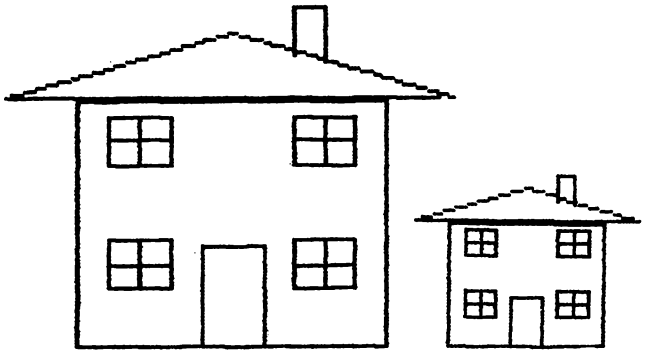


Figure 7.15 Two versions of the HOUSE procedure

Appendix A



List of Commands

While you were working your way through this book, you learned a lot of commands.

To make it easier for you to remember and use them all, I have made this list for you. The list tells what each command does. It tells the short forms for commands that have them. If a command has no shorter form, I've put this mark — by its name under the SHORT column:

COMMAND	SHORT	ACTION
DRAW	—	Puts Logo into the DRAW mode. (DRAW mode homes the turtle and clears the screen.)
FORWARD	FD	Moves the turtle forward as far as you say.
BACK	BK	Moves the turtle backward as far as you say.
RIGHT	RT	Turns the turtle the number of degrees you say to the right.
LEFT	LT	Turns the turtle the number of degrees you say to the left.
HIDETURTLE	HT	Makes the turtle invisible.

COMMAND	SHORT	ACTION
SHOWTURTLE	ST	Returns the picture of the turtle to the screen.
FULLSCREEN	<f5>	Shows the full screen as a drawing board.
TEXTSCREEN	<f1>	Shows a full screen of text.
SPLITSCREEN	<f3>	Screen is part text and part drawing board.
PENUP	PU	Raises the pen so the turtle won't leave any trail when it moves.
PENDOWN	PD	Lowers the pen so the turtle will leave a trail.
PENCOLOR	PC	Allows you to change the color of the trail the turtle leaves.
PENERASE	PC – 1	Allows you to erase lines drawn by mistake.
BACKGROUND	BG	Allows you to change the background color.
REPEAT	—	Repeats the series of commands inside the square brackets [ ] as many times as you say.
TO *NAME*	—	Used with procedure name, enters EDIT mode.
<CTRL>C or <RUN/STOP>	—	Used to define a procedure and exit the EDIT mode.
<CTRL>G	—	Used to exit the EDIT mode without making changes, also used to stop the turtle during a procedure.
↑	—	In DRAW mode, copies the last command you typed.
HOME	—	Homes the turtle without clearing the screen.
CLEARSCREEN	CS	Clears the screen without homing the turtle.

COMMAND	SHORT	ACTION
SAVE "*NAME*	—	Saves the procedures that are in the computer memory onto the work disk.
SAVEPICT:	—	Saves the picture on the screen onto the work disk, with the name you give it.
READ "*NAME*	—	With the name you specify, loads the procedures from the work disk into the computer memory.
READPICT:	—	Loads the picture you name into the computer memory from the work disk.
PRINTPICT:	—	If you have a printer this command lets you print on paper a picture you have saved on your work disk. (This routine is found on the utility disk.)
CATALOG	—	Prints on the screen a listing of everything that is on the work disk.
POTS	—	(Print Out Titles): prints onto the screen a list of all the procedures that you name.
PO *NAME*	—	Prints out onto the screen a list of all the steps in a procedure that you name.
TELL	—	Lets you choose which sprite you want to obey the command you are about to give.
PLAY	—	A routine on the utility disk that lets you teach your turtle to play music.

COMMAND	SHORT	ACTION
WRAP	—	Lets the turtle run off one edge of the screen and come back on the opposite edge. (This is the default setting.)
NOWRAP	—	Stops the turtle from running off the edge of the screen, and also stops the procedure.

Appendix B



Other Reading on Logo

This book is not, and was never meant to be, a complete tutorial for the Logo language.

Its main purpose is to show you how you can have lots of fun while learning about Logo.

I hope you will want to do more reading and studying about Logo. Here is a brief list of other books and magazine articles for you to read.

The book that came with your Commodore 64 Logo is the best and most in-depth work on the subject. It is:

Logo a Language for Learning, the Commodore 64 Logo Tutorial, by Virginia Carter Grammer, E. Paul Goldenberg, and Leigh Klotz, Jr.

Some other worthwhile books that will help you with Logo are:

The Turtle's Sourcebook, by D. Beardon, J. Muller, and C. Martin.

Mindstorms: Children, Computers, and Powerful Ideas, by S. Papert.

Appendix C



Play Time Activity Solutions

Throughout this book there have been Play Time sections. These contained projects for you to solve and have fun with.

In case you have had problems with some of the Play Time activities, I have included this section of hints and answers.

Before you look up the hint or answer, you should make every effort to figure out the answer yourself. The best way to learn is by doing the activity.

If you do need help, don't feel bad. I don't always get the results I expect on my first try either. However, I often learn a lot about how something works when I figure out why my answer is incorrect. You will also discover that some of the results you get, while they may not be what you expect, are interesting by themselves.

CHAPTER 1

1.1 When you finish this one you should be able to move the turtle to any spot on the screen you want.

1.2 Back the turtle up to the bottom edge (don't forget to use the <f5> key) and then move it forward in stages until it disappears over the top edge of the screen. You will find

that it is 254 steps from the bottom of the full screen to the top.

1.3 It is about 404 steps from the bottom left corner of the screen to the top right corner. Back the turtle up to the bottom right corner with the turtle pointing up and then turn the turtle to the RIGHT 47.5 degrees.

1.4
```
FD 25

RT 90

FD 25

RT 90

FD 25

RT 90

FD 25

RT 90
```

1.5
```
PU

LT 135

FD 7.5

RT 135

PD

FD 35

RT 90

FD 35

RT 90

FD 35
```

```
RT 90

FD 35

RT 90
```

1.6 Be sure to use the PU command, then place the turtle in one corner of the screen. Use the PD command and draw your square. Then use the PU command and move the turtle to the next corner. Use the PD command and draw the next square. Keep doing this until you have a square in each corner.

CHAPTER 2

2.1 This one you shouldn't need any help with.

2.2
```
TO ROW

LT 90

REPEAT 4[SQUARE1 FD 25]
```

2.3
```
TO TOWER

REPEAT 4[SQUARE1 FD 25]
```

2.4
```
PU

LT 135

FD 7.5

RT 135

PD

BIGSQUARE
```

2.5 After doing 2.4, type:
```
PU

HOME

PD

TRI
```

2.6
```
TO RECTANGLE

REPEAT 2[FD 25 RT 90 FD 75 RT 90]
```

2.7
```
TO SQUAREX
REPEAT 3[FD 25 RT 135 FD 25 RT 135 FD 25
   RT 90 FD 25 RT 90]
```

2.8
```
TO LEAN

RT 45

REPEAT 4[SQUARE1 FD 25]
```

2.9
```
TO SMALLSQUARE

REPEAT 4[FD 10 RT 90]

TO WINDOW

REPEAT 4[SMALLSQUARE FD 20 RT 90]
```

CHAPTER 3

3.1 The more circles you draw, the easier it will be to control their size.

3.2
```
CIRCLE
```

3.3
```
PU LT 90

FD 15

RT 90

PD

REPEAT 90 FD 2 RT 360/90]
```

3.4 BIGSQUARE

```
PU
RT 90
FD 12
LT 90
FD 25
PD
CIRCLE
```

3.5 REPEAT 45[FD 2 RT 360/90]

3.6 REPEAT 60[FD 2 LT 360/60]

You could also use:

```
RT 180
CIRCLE
```

3.7 REPEAT 90[FD 1 RT 360/90]

REPEAT 90[FD 2 RT 360/90]

REPEAT 90[FD 3 RT 360/90]

3.8 It's fun!!!

3.9 TO SPIN2 :DEGREES :SIDES :LENGTH

```
REPEAT 360/:DEGREES[RT :DEGREES DESIGN :SIDES
        :LENGTH]
```

Try it a few times, and then use SPIN2 360/8 8 50.

If you make :DEGREES 360 divided by the number of sides you get some interesting pictures.

3.10 You already have one, SPIN2. Try SPIN2 10 50 5. Don't forget to use procedures that you have already defined. Don't forget the HT command.

3.11
```
TO TREE

FD 20

LT 90

CIRCLE
```

Or:

```
TO STRAIGHTCURVE

REPEAT 6[FD 15 LT 60 CIRCLE]
```

CHAPTER 4

4.1 The more you play the more you learn. It's also fun to do.

4.2
```
TO RECURWIND :TIMES

IF :TIMES < 1 STOP

REPEAT 4[FD 10 RT 90]

FD 20

RT 90

RECURWIND :TIMES - 1
```

This procedure will work as long as you make :TIMES = 4 when you start.

4.3
```
TO GROW :SIZE :CHANGE

REPEAT 4[FD :SIZE RT 90]

GROW :SIZE + :CHANGE :CHANGE
```

Try this a few times, and let it go for a while.

If you want to keep it from running off the screen add a new first line to limit the size of the sides.

```
TO GROW :SIZE :CHANGE

IF :SIZE > 130 STOP

REPEAT 4[FD :SIZE RT 90]

GROW :SIZE + :CHANGE :CHANGE
```

About the Author

Richard Lambert has been computing for 1½ years and lives in Portland, Oregon.

Index

ANIMAL, 69
Animation, 78
Arithmetic signs, 64

BACK, 22
BACKGROUND, 27
BG, 24
BIGX, 81
BIGY, 81
BK, 24

CATALOG, 43
Circles, 47
CLEARSCREEN, 23
Closed loop, 61
Colon, 51
Color chart, 29
Colors, 27
Commodore keyboard, 7
Condition signs, 65
Conditions, 61
Correcting mistakes, 7, 12
CRSR, 7
CS, 24

<CTRL>C, 31
<CTRL>G, 31, 61
<CTRL>W, 69
Cursor, 7

DEFAULT, 84
Degrees, 13
DELete, 7
Disk name, 9, 42
Division, 37
DRAW, 15, 20

Easy Logo, 70
EDIT mode, 31
ERASEPICT ", 56

<f1>, 24
<f3>, 24
<f5>, 24
FD, 23
Format, 42
Formatting disks, 9
FORWARD, 16
FULLSCREEN, 19

GOODBYE, 43

HIDETURTLE, 48
HOME, 23
HT, 49

id, 9
INPUTS, 52
INSert, 7
INSTANT, 71
<INST DEL>, 7

LEFT, 18
Loading Logo, 10
Logo utility disk, 57, 67
Loops, 60
LT, 24

NOWRAP, 21

Open loop, 61

PC, 24, 28
PC-1, 24
PD, 24
Pencolor, 28
PENDOWN, 22
PENUP, 21
Pictures, 47, 55
PIG, 68
Pig Latin, 67
Pixels, 85
PLAY, 73
PO, 53
POTS, 53
Print out titles, 53
PRINTPICT, 57
Procedures, 29
PU, 24

READ ", 44
READPICT ", 56
Recursion, 60
REPEAT, 33
RIGHT, 17
Right angle, 14
RT, 24

SAVE ", 42
SAVEPICT, 56
Saving your work, 41
Scrolling, 69
Shape files, 80
<SHIFT>, 7
Shortcuts, 23
SHOWTURTLE, 49
SMALLX, 83
SMALLY, 84
SPLITSCREEN, 20
Sprite blocks, 87
Sprite editor, 84
SPRITES
Square brackets, 33
SQUARE DEFINED, 32
ST, 49
Stretching sprites, 81

TELL, 80
TEXTSCREEN, 20
Turtle, 15

Variables, 47, 51

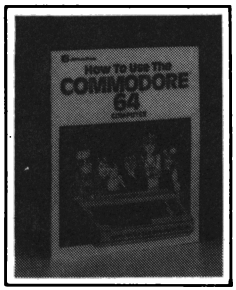
Words, 67
WRAPPING, 19

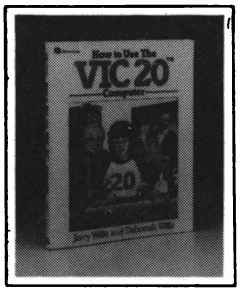
xx, 42

↓, 7
↑, 7, 13, 35
→, 7
←, 7
?, 32
[ ], 33
/, 37, 64
", 43
:, 51
<, 63, 65
+, 64
−, 64
*, 64
=, 65
>, 65
< >, 65
<=, 65
>=, 65

MORE HELPFUL WORDS FOR YOU

from dilithium Press





HOW TO USE THE COMMODORE® 64™

Jerry Willis and Deborrah Willis

ISBN 0-88056-133-6 144 pages/20 illustrations $6.95

HOW TO USE THE VIC 20™ COMPUTER

Jerry Willis and Deborrah Willis

ISBN 0-88056-134-3 184 pages/32 illustrations $6.95

A helpful introduction to your computer and its basic components, this book shows you how to set up or install your computer and get it up and running in no time! Learn how to load and save your own programs as well as programs published in books and magazines.

KEEPTRACK™

File Manager for Personal Computers

Norm Church

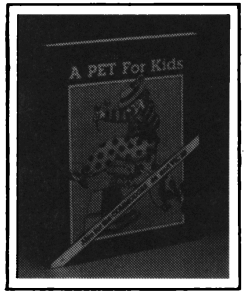
This book and software package turns your Commodore 64 or VIC 20 computer into a personal filing cabinet! It's designed for easy access and will "keep track" of everything from birthdays to tax expenses.

BOOK/SOFTWARE PACKAGES:

ISBN 0-88056-185-8 5¼" disk $29.95
ISBN 0-88056-192-0 cassette $29.95

Software (either disk or cassette) contains programs for Commodore 64 computers on one side, and programs for VIC 20 computers on the other.





A PET FOR KIDS

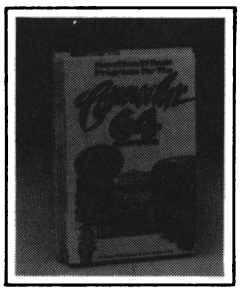
Examples work on Commodore 64 and VIC 20 computers.

Sharon Boren

Here is a fun and instructive approach to teaching kids programming in BASIC and computer operation! Illustrations and examples motivate children ages 8 to 13 to higher learning levels.

ISBN 0-88056-106-8 200 pages $9.95
140 illustrations

Student workbook and teacher's guide available.





MORE THAN 32 BASIC PROGRAMS FOR THE COMMODORE® 64™

Tom Rugg, Phil Feldman and Western Systems Group

BOOK:	ISBN 0-88056-112-2	364 pages/134 illustrations	$19.95
BOOK/SOFTWARE PACKAGES:	ISBN 0-88056-160-7	5¼" disk	$39.95
	ISBN 0-88056-163-1	cassette	$39.95

MORE THAN 32 BASIC PROGRAMS FOR THE VIC 20™ COMPUTER

Tom Rugg, Phil Feldman and Clarence S. Wilson

BOOK:	ISBN 0-88056-161-5	364 pages	$19.95
BOOK/SOFTWARE PACKAGE:	ISBN 0-88056-059-2	cassette	$39.95

Here is a collection of programs for your entire family. It's chock-full of programs with practical applications, educational uses, games, and graphics too! Type in programs from the book or use ready-to-run programs provided in the book/software package.

BRAIN FOOD — Our free catalog listing over 150 microcomputer books covering software, hardware, business applications, general computer literacy and programming languages.

dilithium Press books are available at your local bookstore or computer store. If there is not a bookstore or computer store in your area, charge your order on VISA or MC by calling our toll-free number, **(800) 547-1842.**

Send to: dilithium Press, P.O. Box E, Beaverton, OR 97075. Please send me the book(s) I have checked. I understand that if I'm not satisfied I can return the book(s) within 10 days for full and prompt refund.

☐ HOW TO USE THE COMMODORE 64
☐ HOW TO USE THE VIC 20
KEEPTRACK
Book/Software Packages:
☐ Disk: ☐ Cassette
A PET FOR KIDS
☐ Book ☐ Workbook ☐ Teacher's Guide

MORE THAN 32 BASIC PROGRAMS FOR THE VIC 20
☐ Book
Book/Software Packages:
☐ Disk ☐ Cassette

MORE THAN 32 BASIC PROGRAMS FOR THE COMMODORE 64
☐ Book
Book/Software Packages:
☐ Disk ☐ Cassette

☐ Check enclosed $_____ payable to dilithium Press

Please charge my Visa ☐ MC ☐

#_____ Exp. Date_____

Name_____

Address_____

Signature_____

City, State, Zip_____

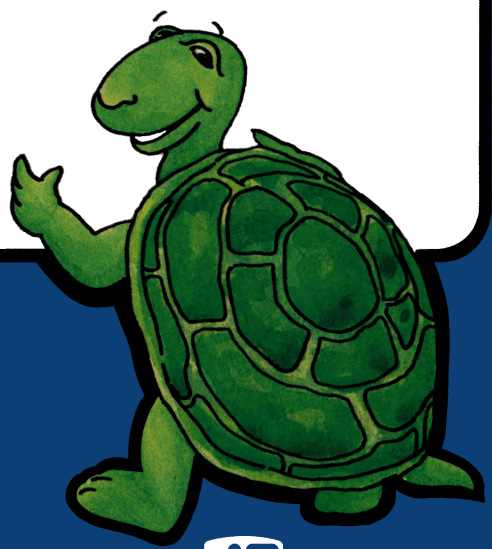
☐ Send me your free catalog, **BRAIN FOOD.**

prices are subject to change

All children enjoy playing with turtles, but not all turtles help kids learn! Written for children ages 7 and up, **Logo and More for the Commodore® 64™** creatively introduces a fun turtle mascot that teaches them how to use the programming language Logo on their Commodore 64. Using a hands-on approach, it teaches kids the basics of Logo by incorporating over 30 educational playtime activities. Fully illustrated, this book encourages children to write programs that draw colorful designs and play lively music. **Logo and More for the Commodore® 64™**:

- Shows you how to turn on the Commodore 64.
- Explains how to load Logo.
- Contains programs you can type in yourself that teach you Logo.
- Tells you how to write your own procedures.
- Develops problem-solving skills.
- Offers a solid programming foundation.

Logo and More for the Commodore® 64™ supplies all the commands a child needs to get started with Logo. And with this little turtle, children will adore learning "Logo and More" on their Commodore 64!



ISBN 0-88056-348-6

0 45078 00795 56348

dilithium Press
EDUCATIONAL



This was brought to you

from the archives of

http://retro-commodore.eu