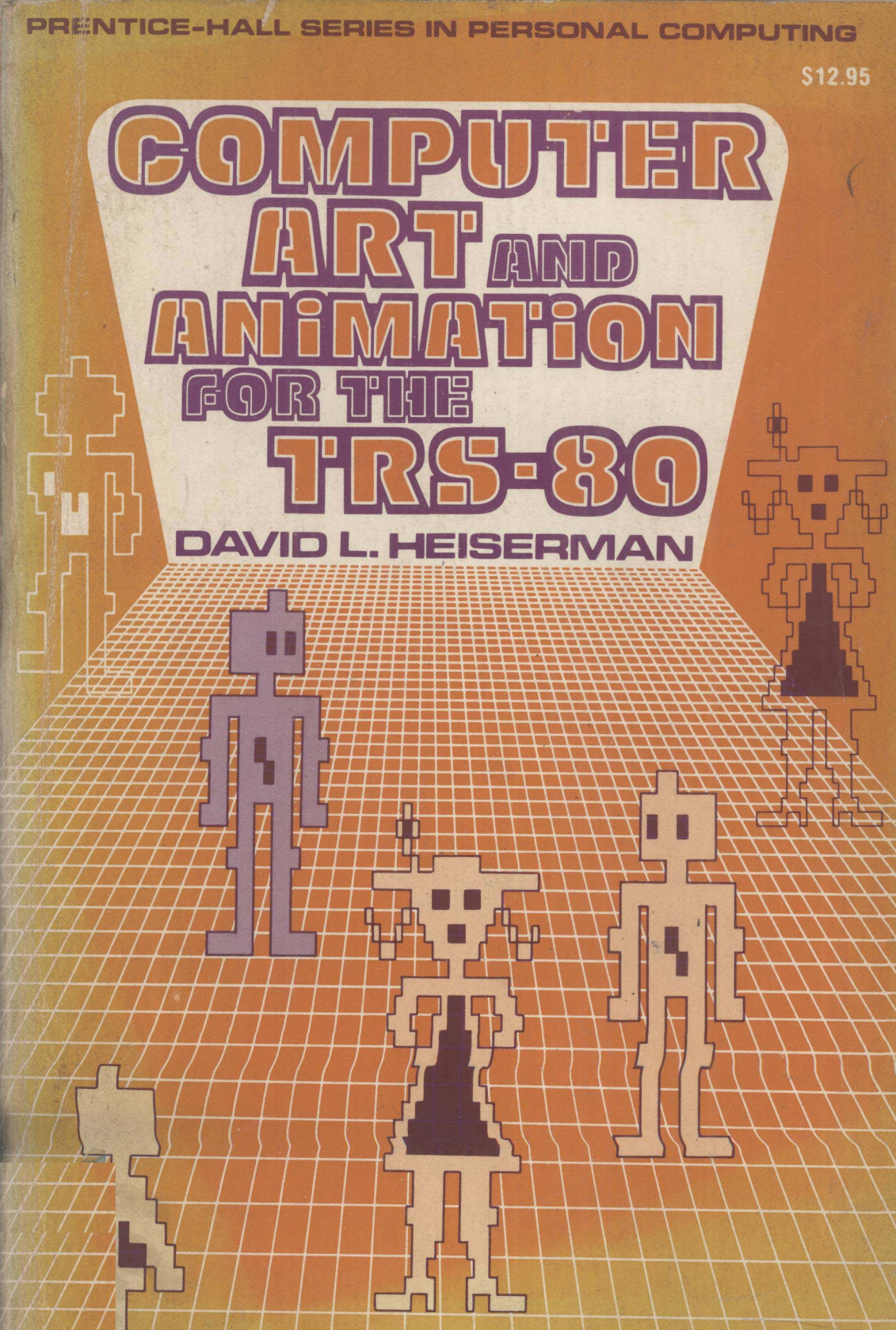
PRENTICE-HALL SERIES IN PERSONAL COMPUTING

$12.95

COMPUTER ART AND ANIMATION FOR THE TRS-80

DAVID L. HEISERMAN



Computer Art and Animation for the TRS-80

David L. Heiserman

PRENTICE-HALL, INC. Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

Heiserman, David L., (date)
Computer art and animation for the TRS-80.

(Prentice-Hall series in personal computing)
Includes index.
1. Computer graphics. 2. Computer art.
3. Computer animation. 4. TRS-80 (Computer)—
Programming. I. Title. II. Series.
T385.H44 1982 001.55'3 82-16592
ISBN 0-13-164749-0
ISBN 0-13-164731-8 (pbk.)

Editorial/Production Supervision
and Interior Design: *Lynn S. Frankel*
Cover Design: *Photo Plus Art*
Manufacturing Buyer: *Gordon Osbourne*

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-164749-0 {CASE}
ISBN 0-13-164731-8 {PAPER}

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Editora Prentice-Hall do Brazil, Ltda., *Rio de Janeiro*
Prentice-Hall of Canada, Ltd., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

Preface

Computer graphics plays a vital role in the successful application of home computer systems. Complex ideas are often better expressed in terms of animated graphics sequences, columns of dull data can be more meaningful when backed up with graphs or moving images, and, of course, computer games can be far more exciting when they are built around graphic razzle-dazzle.

The literature usually supplied with a home computer system, however, rarely reflects the full potential of its graphics system. The subject is often treated as a novelty to be exploited only by those who care to spend the time figuring things out for themselves.

This book treats TRS-80 computer graphics and animation as a topic worthy of considerable study and experimentation. The discussions and examples should help a programmer develop a sense of confidence in developing programs that employ some graphics features. The programs may be almost wholly graphic, serving as art for its own sake, or they may

serve as a small part of a larger program that has a more practical purpose.

Whatever one's motivation might be, the study of computer graphics and animation can reap highly satisfying rewards. It's a matter of giving it the serious attention it requires. This book is a step in that direction.

DAVID L. HEISERMAN

CONTENTS

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 8

CHAPTER 9

A FIRST LOOK AT TRS-80 ANIMATION

CHAPTER 10

COMPLEX-FIGURE ANIMATION

CHAPTER 11

MOVING FIGURES FROM PLACE TO PLACE

CHAPTER 12

KEYBOARD CONTROL AND CONTACT SENSING

1 Setting the Stage

If your concept of computer art is limited to doodling complicated mathematical functions on a CRT, your understanding of the matter is about 20 years behind the times. Computer art, or graphics, is now a legitimate branch of computer science, and it certainly hasn't gained that stature by limiting itself to plotting purely abstract figures that a child can create with a simple drawing toy.

If you think it takes a great deal of expensive, special equipment to create computer animated sequences, you are in for a pleasant surprise. Cost and sophistication are relative terms, of course; assuming you already own a home computer system, the additional cost is just about nil, and there is no need for building or buying any additional equipment.

That is not to say that it is easy to produce satisfying computer art and animations. It is hard work, and it requires some learning and discipline. The purpose of this book is to point you toward the techniques that are most likely to produce the results you want. It is up to you to supply the creativity and hard work.

THE REQUIRED EQUIPMENT

All of the discussions and examples cited in this book are oriented toward 16k, TRS-80 home computer systems; either the Model 1 with Level II BASIC, or the Model 3. It is assumed you will be using a cassette tape system for saving programs, although a disk-based TRS-80 system works equally well.

Aside from the standard drawing tools—pencils, a drawing compass, protractor, and a good supply of erasers— the only auxiliary materials of special importance are some clean cassette tapes and a couple of pads of Video Programming Worksheets (Radio Shack catalog number 28-2105).

PROGRAMMING PREREQUISITES

All of the discussions are based on the BASIC programming language. A knowledge of BASIC is absolutely essential for getting any real benefit from the work presented here.

The graphics techniques featured throughout the book are slanted toward the application of string variables, dimensioned arrays, and DATA listings. If you happen to be weak in your understanding of any of those families of BASIC statements (as many home computer programmers are), you should take some time to polish your understanding as you work into the first few chapters.

A WORD ABOUT LOW-RESOLUTION GRAPHICS

The TRS-80's graphics system is usually classified as a low-resolution system. It is virtually impossible, for instance, to draw a circle that doesn't have some geometric distortion and a rough, step-like appearance in most places. Therefore, unless the desired figure happens to be a rectangle that is situated so that its sides are parallel to the edges of the CRT screen, one has to settle for something less than a perfect rendition of a desired image. However, that ought to be taken as challenge rather than a point of discouragement.

The human brain is a marvelous perceptive organ, but it can be tricked. It is possible to fool the brain into thinking it is seeing something that isn't there; in the context of producing satisfying low-resolution pictures, that means it is possible to fool the brain into ignoring imperfections. With creative application of low-resolution graphics, a viewer can perceive the real essence of a figure or animation sequence without being bothered by geometric distortions that are inevitably present in it.

Viewed objectively, a TRS-80 image of a girl standing in the middle

of the screen can look terrible. The figure is necessarily made up of sharp right angles, relatively large rectangles of black and white, with none of the smooth curves a traditional artist would like to use. However, if you take time and care in preparing the figure, if you take the trouble to introduce some novel and interesting elements, the brain will attempt to interpret the image as something rather nice and meaningful. It will attempt to view the image as the artist intended, reaching for the essence of the image rather than dwelling on its objective appearance. Furthermore, the effect is enhanced when the image is animated in real time.

Along the same lines, the TRS-80 systems featured here are limited to black and white drawings. There are no shades of gray. That, too, poses no problem when one consciously introduces strokes of black and white that merely represent shades of gray. So what if the shadow cast by a figure is white instead of black or gray? The visual impression—especially if animated—can still be quite effective. It is a matter of taking the trouble to try it.

HOW TO USE THIS BOOK

Take the book one step at a time. A number of special graphics techniques are developed over a number of chapters; attempting to jump into the middle of the book without studying the preceding material is courting disaster. So start from the beginning and move along as rapidly as you wish, making sure you are grasping the essence of the material as you go along.

There are a good many specific programming examples suggested through these chapters. They are intended to be just that—examples. This is not a collection of finished programs that stand on their own merits. It is up to you to develop such programs yourself. Try the examples in order to convince yourself that the topic at hand really fulfills its purpose. Study the programs themselves to make sure you understand exactly how they work and what the function of every statement is. I hope you will find some of the specific program listings interesting and even a little amusing, but you will be selling yourself short if you make no attempt to generate programs of your own as you go through the book.

2 The TRS-80 Graphic Characters

The graphic characters available on the TRS-80 are the elementary components of any image that can be drawn on the CRT screen. No picture can be created without them, so a discussion of those characters is a logical starting point for any presentation of TRS-80 graphics and animation.

An imaginative, creative selection of graphic elements goes a long way toward creating satisfying images on the screen. Anything less than a thorough appreciation of all those characters—exactly how they look and how to get them onto the screen—is bound to diminish the quality of the final works.

Even if you have worked with the TRS-80 graphic characters before, you will do well to study this chapter because it goes into a finer level of detail than does most of the standard literature on the subject. It is this eye for detail that promises to open new horizons for graphic achievement.

This chapter is not concerned with putting the graphic characters into a desired place on the screen, nor with stringing characters to create larger and more complex figures. That begins in Chapter 3. For now, use

the programs merely to view the characters they place onto the screen. It's not necessary yet to figure out how the programs work. You will have an abundance of program analysis later on.

CHARACTER SPACES

For our immediate purposes, the TRS-80's graphics are divided into two categories: the alphanumeric characters, including punctuation marks and special symbols, and the unique TRS-80 graphic elements. There are 64 of each kind, and in this chapter you will get to know all 128 of them rather well.

The CRT screen has a rectangular *graphics field.* That is the working space for all the characters, leaving a black border around the edges where nothing can be printed. That character field is divided into 1024 separate character locations, on each horizontal line of text or graphics, and 16 lines running from top to bottom of the field. Any character, alphanumeric or special graphic, has to fit into one of those character locations in the field. There are no in-between places.

Each of those 1024 character locations is further divided into 6 smaller segments called *pixels.* As indicated in Fig. 2-1, the pixels are arranged in a 2 × 3 format.

The smallest graphic that can be plotted is one that fills out just one pixel. Some of the keyboard punctuation marks (comma, period, and apostrophe, for instance) fill only a portion of one pixel, but they cannot be drawn without excluding any other graphics from the rest of that character space. Since the pixels are the elementary building blocks for screen graphics, they are worthy of closer study.

Obviously the pixels are not square. Using simple eyeball reckoning, and perhaps a natural desire to make things as simple as possible, one might assume that each pixel is about twice as tall as it is wide. But that is not the case. The relative dimensions of each pixel is actually 1 × ⅓: it is 2⅓ times taller than wide.

An important implication of the rectangular pixel shape is that lines drawn vertically can be much narrower than those drawn horizontally. Horizontal resolution, in other words, is far better than vertical resolution. If you have played with some SET and RESET graphics, you have doubtless noticed that fact. It is something that TRS-80 graphics artists must contend with constantly.

So each pixel in a graphic character space has relative dimensions of 1 × ⅓. Since the pixels are arranged in a 2 × 3 pattern within each character space, it follows that the relative dimensions of each character space are 2 × 7. A character space is thus 3½ times taller than it is wide. When it comes to printing graphics from the alphanumeric character

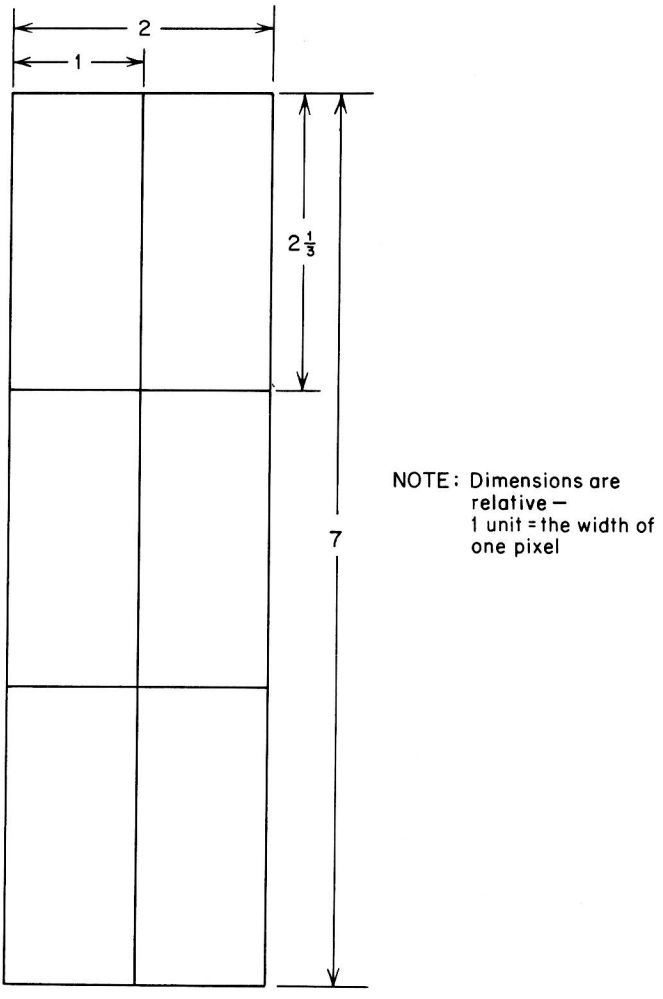


FIGURE 2-1 A TRS-80 character space showing the relative dimensions of that space and its pixels

generator, each character will dominate a 2 × 7 space in the field. The space in that case will always be black and the character itself will be white. No other graphic can be slipped into that same space as long as the character is there.

Finally, since there are 64 character spaces in each field line and 16 lines in the field, the relative dimensions of the field is 8 × 7. The field is slightly wider than it is tall.

No doubt matters would be much simpler if everything were square—pixels, character spaces, and the field. Things aren't that simple,