

The TUTOR Language

Bruce Arne Sherwood

To my Mother, who saw me through!

Love,

Bruce

FUND of PLATO PROY
DAVIS

The TUTOR Language

The TUTOR Language

Bruce Arne Sherwood
*Computer-based Education Research Laboratory
and Department of Physics
University of Illinois
Urbana, Illinois*

CONTROL DATA
EDUCATION COMPANY

 a service of
CONTROL DATA CORPORATION

© 1977 by Bruce Arne Sherwood.

All rights reserved. No part of this material may be reproduced by any means without permission in writing from the publisher and the author.

ISBN: 0-918852-00-5

Library of Congress Catalog Card Number: 77-77589

Printed in the United States of America

Preface

The PLATO IV computer-based education system was developed in the Computer-based Education Research Laboratory (CERL) of the University of Illinois, Urbana. PLATO IV is the result of 15 years of research and development effort led by Donald Bitzer, director of CERL. The University of Illinois system presently links 950 graphical-display terminals to a large Control Data Corporation computer in Urbana. Some of these terminals are located as far away as San Diego and Washington, D.C. Additional PLATO systems with their own complements of terminals are located elsewhere in the United States. Students are individually tutored at terminals by interacting with PLATO lesson materials created by teachers. There are over 4,000 hours of PLATO lessons already available. These lessons span a wide range of subject areas and are used by students in elementary schools, community colleges, military training bases, universities, and commercial training programs. Authors of lesson materials are teachers who use the TUTOR language to tell PLATO how to interact with students on an individual basis. This book explains the TUTOR language in detail and is intended to help authors write quality lesson materials.

In 1967, Paul Tenczar (then a graduate student in zoology) concluded that existing methods of creating computer-based lesson material on the earlier PLATO III system were unnecessarily difficult. As a result he originated the TUTOR language. There followed a rapid increase in the number of authors and in the number and degree of sophistication of

the lessons they wrote. This active author community in turn spurred the continual development and refinement of TUTOR by requesting additional needed features. In 1970, CERL began implementing the PLATO IV system, which afforded a rare opportunity to take stock of the evolution of TUTOR up to that point and make a fresh start. Many useful simplifications were made, and many important features were added. The growth of PLATO IV into a continental network brought together an ever-wider spectrum of authors through the rich interpersonal communications facilities available on PLATO, and the suggestions and criticisms from these authors contributed to the present form of the TUTOR language. Also of great importance has been the large number of students who have used PLATO lessons, and whose experiences have influenced the development of TUTOR to meet their needs. The TUTOR language described in this book is, therefore, based on heavy use-testing.

In the earliest phase Paul Tenczar and Richard Blomme were mainly responsible for TUTOR development. Since then, many people have been involved, some as full-time CERL staff members and some as high school, undergraduate, or graduate students. It is impossible to adequately acknowledge the various contributions, and difficult even to list all of those who have played a major role, but an attempt should be made. Paul Tenczar is head of TUTOR development. Full-time people have included David Andersen, Richard Blomme, John Carstedt (Control Data), Ruth Chabay, Christopher Fugitt, Don Lee, Robert Rader, Donald Shirer, Michael Walker, and this author. They have been assisted by James Parry and Masako Secrest, and by Doug Brown, David Frankel, Sherwin Gooch, David Kopf, Kim Mast, Phil Mast, Marshall Midden, Louis Steinberg, Larry White, and David Woolley. William Golden has also provided useful advice.

All of these people have been involved mainly with “software”, the programming of the PLATO computer in such a way as to permit authors and students to write and use computer-based lessons. Of equal importance to the technical success of PLATO are the CERL scientists, engineers, and technicians who invented, designed, and implemented the unique terminals and telecommunications devices (“hardware”) which form the PLATO educational network. CERL personnel who have been heavily involved in hardware development include Donald Bitzer, Jack Stifle, Fred Ebeling, Michael Johnson, Roger Johnson, Frank Propst, Dominic Skaperdas, Gene Slottow, and Paul Tucker.

The latter part of Chapter 1 is adapted from a PLATO III document, “The TUTOR Manual”, by R. A. Avner and P. Tenczar.

I thank Elaine Avner and Jeanne Weiner for editorial assistance, Sheila Knisley for typing, and Stanley Smith for photographic work. I appreciate the encouragement William Golden gave me to finish the task.

Contents

1	Introduction	1
	How to Use This Book	1
	Sample PLATO Lessons	3
	The PLATO Keyboard	8
	Basic Aspects of TUTOR	13
2	More on Creating Displays	23
	Coarse Grid and Fine Grid	23
	The -box-, -vector-, and -circle- Commands	25
	Large-size Writing: -size- and -rotate-	26
	Animations (Moving Displays): -erase-	
	and -pause-	28
	-pause-, -time-, and -catchup-	30
	The -mode- Command	33
	Automated Display Generation	35
3	Building Your Own Tools: The -do- Command	39
4	Doing Calculations in TUTOR	43
	Giving Names to Variables: -define-	47
	Repeated Operations: The Iterative -do-	49
	Showing the Value of a Variable	51
	Passing Arguments to Subroutines	53

Contents

5	Sequencing of Units Within a Lesson	59
	Summary of Sequencing Commands	69
	The -helpop- Command: "Help on Page"	72
	The -imain- Command	73
6	Conditional Commands	77
	Logical Expressions	80
	The Conditional -write- Command (-writec-)	82
	The Conditional -calc- Commands: -calcc- and -calcs-	84
	The Conditional -mode- Command	85
	The -goto- Command	85
	The Conditional Iterative -do-	90
	The -if- and -else- Commands	91
7	Judging Student Responses	95
	Student Specification of Numerical Parameters	101
	Student Specification of Non-Numerical Parameters	104
	Difference Between Numeric and Alphabetic Information	105
	More On -answer- and -wrong- (Including -list- and -specs-)	106
	Building Dialogs With -concept- and -vocab-	111
	Numbering Vocabulary Words	117
	The -judge- Command	118
	Finding Key Words: The -match- and -storen- Commands	123
	Numerical and Algebraic Judging: -ansv- and -wrongv-	126
	Handling Scientific Units: -ansu-, -wrongu-, and -storeu-	133
	The -exact- and -exactc- Commands	136
	The -answerc- Command: A Language Drill	137
	Summary	139
8	More About Judging	141
	Stages in Processing the -arrow- Command	141
	Repeated Execution of -join-	142

Judging Commands Terminate Regular State	144
The <code>-goto-</code> is a Regular Command	146
Interactions of <code>-arrow-</code> with <code>-size-</code> , <code>-rotate-</code> , <code>-long-</code> , <code>-jkey-</code> , and <code>-copy-</code>	149
Applications of <code>-jkey-</code> and <code>-ans-</code>	151
Modifying the Response: <code>-bump-</code> and <code>-put-</code>	156
Manipulating Character Strings	159
Catching Every Key: <code>-pause-</code> , <code>-keytype-</code> , and <code>-group-</code>	164
Touching the Screen	168
Summary	169
9 Additional Display Features	171
More on the <code>-write-</code> Command	171
Extensions to the Basic Character Set	175
The “initial entry unit” (<code>ieu</code>)	177
Smooth Animations Using Special Characters	178
Creating a New Character Set	179
Micro Tables	181
The Graphing Commands: Plotting Graphs with Scaling and Labeling	182
Summary of Line-drawing Commands: <code>-draw-</code> , <code>-gdraw-</code> , <code>-rdraw-</code>	185
The <code>-window-</code> Command	190
More on Erasing: The <code>-eraseu-</code> Command	192
Keeping Things on the Screen: “inhibit erase”	196
Interaction of “inhibit erase” with <code>-restart-</code>	199
The <code>-char-</code> and <code>-plot-</code> Commands	199
The <code>-dot-</code> Command	200
10 Additional Calculation Topics	201
Defining Your Own Functions	202
Arrays	204
Segmented Variables	207
Branching Within a Unit: <code>-branch-</code> and <code>-doto-</code>	212
Array Operations	214
<i>Integer Variables and Bit Manipulation</i>	217
Byte Manipulation	229

Contents

	Vertical Segments	230
	Alphanumeric to Numeric: The -compute- Command	231
	The -find- Command	235
	The -exit- Command	236
11	Manipulating Data Bases	237
	The -common- Command	237
	The Swapping Process	240
	Common Variables and the Swapping Process	243
	The -storage- Command	246
	Using -datasets-	248
	Sorting Lists	248
12	Miscellany	249
	Other Terminal Capabilities	249
	Student Response Data	251
	Additional Tools for Teaching Foreign Languages	252
	Routers and -jumpout-	254
	Instructor Mode	255
	Special “terms”	255
	APPENDICES	257
	Appendix A. Where to Get Further Information	258
	Appendix B. List of TUTOR Commands Additional TUTOR Commands Not Discussed in This Book	259 260
	Appendix C. List of Built-in -Calc- Functions	261
	System Variables	262

