TUTOR Guides and Articles

The TUTOR Language

The Articles in this work are adapted from:

The TUTOR Language by Bruce Arne Sherwood

Computer-based Education Research Laboratory and Department of Physics University of Illinois
Urbana, Illinois Copyright © 1977 by Bruce Arne Sherwood.

ISBN: 0-918852-00-5 Library of Congress Catalog Card Number: 77-77589

Reproduction and Repurpose granted by Author.

- Preface
  1. Introduction
    1. How to Use This Book
    2. Sample Plato Lessons
    3. The PLATO Keyboard
    4. Basic Aspects of TUTOR
  2. More on Creating Displays
    1. Coarse Grid and Fine Grid
    2. The -box-, -vector-, and -circle- Commands
    3. Large-size Writing: -size- and -rotate-
    4. Animations (Moving Displays): -erase and -pause-
    5. -pause-, -time-, and -catchup-
    6. The -mode- Command
    7. Automated Display Generation
  3. Building Your Own Tools: The -do- Command
  4. Doing Calculations in TUTOR
    1. Giving Names to Variables: -define-
    2. Repeated Operations: The Iterative -do-
    3. Showing the Value of a Variable
    4. Passing Arguments to Subroutines
  5. Sequencing of Units Within A Lesson
    1. Summary of Sequencing Commands
    2. The -helpop- Command: "Help on Page"
    3. The -imain- Command
  6. Conditional Commands
    1. Logical Expressions
    2. The Conditional -write- Command (-writec-)
    3. The Conditional -calc- Commands: -calcc- and -calcs-
    4. The Conditional -mode- Command
    5. The -goto- Command
    6. The Conditional Iterative -do-
    7. The -if- and -else- Commands
  7. Judging Student Responses
1. Student Specification of Numerical Parameters
2. Student Specification of Non-Numerical Parameters
3. Building Dialogs With -concept- and -vocabs-
4. Numbering Vocabulary Words
5. The -judge- Command
6. Finding Key Words: The -match- and -storen- Commands
7. Numerical and Algebraic Judging: -ansv- and -wrongv-
8. Handling Scientific Units: -ansu-, -wrongu-, and -storeu-
9. The -exact- and -exactc- Commands
10. The -answerc- Command: A Language Drill
11. Summary

8. More About Judging
   1. Repeated Execution of -join-
   2. The -goto- is a Regular Command
   3. Interactions of -arrow- with -size-, -rotate-, -long-, -jkey-, and -copy-
   4. Applications of -jkey- and -ans-
   5. Modifying the Response: -bump- and -put-
   6. Manipulating Character Strings
   7. Catching Every Key: -pause-, -keytype-, and -group-
   8. Touching the Screen
   9. Summary

9. Additional Display Features
   1. Additional Display Features
   2. The "initial entry unit" (ieu)
   3. Smooth Animations Using Special Characters
   4. Creating a New Character Set
   5. Micro Tables
   6. The Graphing Commands: Plotting Graphs with Scaling and Labeling
   7. Summary of Line-drawing Commands: -draw-, -gdraw-, -rdraw-
   8. The -window- Command
10. Keeping Things on the Screen: "inhibit erase"
   11. Interaction of "inhibit erase" with -restart-
   12. The -char- and -plot- Commands
   13. The -dot- Command

10. Additional Calculation Topics
   1. Defining Your Own Functions
   2. Segmented Variables
   3. Branching Within a Unit: -branch- and -doto-
   4. Array Operations
   5. Integer Variables and Bit Manipulation
   6. Byte Manipulation
   7. Vertical Segments
   8. Alphanumeric to Numeric: The -compute- Command
   9. The -find- Command
   10. The -exit- Command

11. Manipulating Data Bases
   1. The -common- Command
   2. The Swapping Process
   3. The -storage- Command
   4. Using "datasets"
5. Sorting Lists
12. Miscellany

Appendices

1. Where to Get Further Information
2. List of TUTOR Commands
3. List of Built-in -calc- Functions
   1. System Variables
   2. Alphabetical List of Commands

Alphabetical index

- Additional Calculation Topics
- Additional Display Features
- Alphanumeric to Numeric: The -compute- Command
- Appendix A: Where to Get Further Information
- Appendix B: List of TUTOR Commands
- Appendix C: List of Built-in -calc- Functions
- Applications of -jkey- and -ans-
- Array Operations
- Basic Aspects of TUTOR
- Branching Within a Unit: -branch- and -doto-
- Building Dialogs With -concept- and -vocabs-
- Building Your Own Tools: The -do- Command
- Byte Manipulation
- Catching Every Key: -pause-, -keytype-, and -group-
- Conditional Commands
- Creating a New Character Set
- Doing Calculations in TUTOR
- Finding Key Words: The -match- and -storen- Commands
- Giving Names to Variables: -define-
- Handling Scientific Units: -ansu-, -wrongu-, and -storeu-
- Integer Variables and Bit Manipulation
- Interactions of -arrow- with -size-, -rotate-, -long-, -jkey-, and -copy-
- Introduction
• Judging Student Responses
• Keeping Things on the Screen: "inhibit erase"
• Logical Expressions
• Manipulating Character Strings
• Manipulating Data Bases
• Micro Tables
• Miscellany
• Modifying the Response: -bump- and -put-
• More About Judging
• More on Creating Displays
• More on Erasing: The -eraseu- Command
• Numbering Vocabulary Words
• Numerical and Algebraic Judging: -ansv- and -wrongv-
• Passing Arguments to Subroutines
• Preface
• Repeated Execution of -join-
• Repeated Operations: The Iterative -do-
• Routers and -jumpout-
• Segmented Variables
• Sequencing of Units Within a Lesson
• Showing the Value of a Variable
• Smooth Animations Using Special Characters
• Student Specification of Non-Numerical Parameters
• Student Specification of Numerical Parameters
• Summary of Line-drawing Commands: -draw-, -gdraw-, -rdraw-
• Summary of Sequencing Commands
• The "initial entry unit" (ieu)
• The -answerc- Command: A Language Drill
• The -exact- and -exactc- Commands
• The -exit- Command
• The -find- Command
• The -goto- Command
• The -goto- is a Regular Command
• The -helpop- Command: "Help on Page"
• The -if- and -else- Commands
• The -imain- Command
• The -judge- Command
• The -storage- Command
• The -window- Command
• The Conditional -calc- Commands: -calcc- and -calcs-
• The Conditional -mode- Command
• The Conditional -write- Command (-writec-)
• The Conditional Iterative -do-
• The Graphing Commands: Plotting Graphs with Scaling and Labeling
• The PLATO Keyboard
• The Swapping Process
• Touching the Screen
• Using "datasets"
• Vertical Segments
PLATO Project

The PLATO System PLATO (Programmed Logic for Automatic Teaching Operations) was the first generalized computer-assisted instruction system. Starting in 1960, it ran on the University of Illinois' ILLIAC I computer. By the late 1970s, it supported several thousand graphics terminals distributed worldwide, running on nearly a dozen different networked mainframe computers. Many modern concepts in multi-user computing were originally developed on PLATO, including forums, message boards, online testing, e-mail, chat rooms, picture languages, instant messaging, remote screen sharing, and multiplayer video games.

PLATO was designed and built by the University of Illinois and functioned for four decades, offering coursework (elementary through university) to UIUC students, local schools, and other universities. Courses were taught in a range of subjects, including Latin, chemistry, education, music, and primary mathematics. The system included a number of features useful for pedagogy, including text overlaying graphics, contextual assessment of free-text answers, depending on the inclusion of keywords, and feedback designed to respond to alternative answers.

Rights to market PLATO as a commercial product were licensed by Control Data Corporation (CDC), the manufacturer on whose mainframe computers the PLATO IV system was built. CDC President William Norris planned to make PLATO a force in the computer world, but found that marketing the system was not as easy as hoped. PLATO nevertheless built a strong following in certain markets, and the last production PLATO system did not shut down until 2006, coincidentally just a month after Norris died.

Fast forward to today - a conservation effort was undertaken by a group grass-roots enthusiasts who worked to instantiate the last functioning version of what was the PLATO system (CYBIS). Use of the CYBIS system was granted for non-commercial purposes to the hobbyist community and lives on in the form of Cyber1.Org.

Cyber1 is the name for the mainframe-based CYBIS system. To those familiar with PLATO, CYBIS, or early NovaNET, cyber1 will feel like coming home again. Cyber1 runs on top of NOS, the CDC mainframe operating system, generously contributed by BT Consulting & Systems Integration Services (formerly Syntegra). NOS in turn runs on top of DtCyber (watch out, this is a link to a .pdf), a software emulation of a CDC Cyber mainframe, created by Tom Hunter.

Cyber1.org is a group of people dedicated to the preservation of the world's first computer-based community, PLATO.

PLATO is a computer-based educational system created at the University of Illinois Control Systems Laboratory. The idea was first discussed at the University in 1959, in a long series of meetings led by Chalmers Sherwin. At these meetings it was concluded that computer-based education should not be pursued. However, the director at the time, Daniel Alpert, got together with Donald Bitzer to see if Don could quickly come up with a prototype that could serve as proof-of-concept. This prototype, running on an Illiac-I, became PLATO. The project was subsequently funded in 1960 by government money from the Joint Services Program. The lab grew and became the Computer-based Education Research Laboratory (CERL). PLATO eventually spawned a variety of commercial ventures, starting in 1975 with Control Data Corporation (CDC), a mainframe computer manufacturer founded in 1957.

The complicated histories of the various branches of the PLATO tree are beyond the scope of this...
One of the branches became CYBIS, a product of VCampus; another branch is PLATO Learning, the company that retains the right to the PLATO name; a third became NovaNET, now owned by Pearson Education. Our system, cyber1, is a branch off of the CYBIS tree.

The long list of enthusiasts lives in the group “ControlFreaks” (former employees, affiliates and users of the Control Data Corporation). Among them are those who receive credit for the work that is now, Cyber1:

**Credits**

<table>
<thead>
<tr>
<th>Who</th>
<th>For What</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Donald Bitzer</td>
<td>Leading the PLATO Project at the University of Illinois and creation of the Plasma Panel whose development benefits the display devices and the lives of everyone around the world</td>
</tr>
<tr>
<td>Tom Hunter and ControlFreaks</td>
<td>Mainframe emulation of the CDC Cyber system upon which PLATO runs and the wealth of knowledge represented in the group of practitioners</td>
</tr>
<tr>
<td>British Telecom (BT)</td>
<td>Support of, and limited release of the NOS 2.8.7 Operating System</td>
</tr>
<tr>
<td>Syntegra (Thomas Kennedy, Jim Kubiak, Joann Swoboda)</td>
<td>Responsible for creating the original Concession under which the original revival of PLATO/CYBIS could be accomplished</td>
</tr>
<tr>
<td>Telswitch (Aaron Woolfson)</td>
<td>Authentic terminal restoration and PLATO keyboard enthusiast whose work continues to be on display in the Computer History Museum</td>
</tr>
<tr>
<td>Paul Koning, Dale Sinder, Joe Stanton, other community contributors</td>
<td>Construction of the PLATO V Terminal Emulator (it is open source)</td>
</tr>
<tr>
<td>Nat Kannan (VCampus)</td>
<td>Release of CYBIS to the “Hobbyist” community for preservation</td>
</tr>
<tr>
<td>Mike Cochran, Bill Galcher, Paul Koning, Steve Peltz, Paul Resch, Joe Stanton, Steve Williams</td>
<td>Construction, maintenance, consultation, and general knowledge which has been extraordinarily hard to properly persist - but this CODEX will attempt to preserve a distillation of the operational, functional, and technical knowledge necessary to run the CYBIS Release1 Distribution</td>
</tr>
</tbody>
</table>