The Great Logo Atlas

Search

Login

Last update:
December 09. 2008 22:31:15

TI Logo

Synopsis

Logo was created in 1968. Most Logo work during the 1970s was conducted using large research computer systems. One of the very first commercial versions of Logo for a microcomputer was a coordination of efforts between MIT and Texas Instruments in 1979.

Ben Yates holds a TI Logo with a copyright date of 1981 and TI Logo II with a copyright date of 1983.

According to "The Best of 99'er - Volume 1" (ISBN 0-933094-11-6), p 100 "The Lamplighter LOGO Project":

"Midway through the fall semester of 1979, several early prototypes of TI LOGO were tested at Lamplighter and revised by the MIT LOGO personnel in consultation with Lamplighter and Texas Instruments. In January 1980, the pace of computing at Lamplighter accelerated as an updated version of TI LOGO was implemented on the TI prototypes..."

And from the resident TI Historian, Bill Gaskill, who maintains a TI computer timeline, comes this information:

....From the April 1981 Timeline: TI LOGO (PHM 3040) is introduced on April 17. It is only made available to qualified school districts.

The first advertisement I have for TI LOGO can be found on the back cover of 99er Magazine, V1N5. From the content of the magazine I have 'guessed' that V1N5 had a February 1982 publish date.

The purpose of this page is to document the versions of Logo available for the TI-99/4 computer, *TI Logo* and *TI Logo II*. Within this document, whenever TI Logo is discussed, you can assume that it applies to both TI Logo and TI Logo II.

System Requirements

First, let's go over the system requirements for Logo on the 99/4. These are the Logo command module, 99/4 or 99/4A console, 32k (yes, that is thirty-two kilobytes) of RAM, monitor or TV, and a storage medium. The storage medium may be cassette recorder or disk drives.

Logo Features

Logo has the following features:

- Logo is a *procedural* language.
- Logo is an *interactive* language.
- Logo's data structures include numbers and characters strings, but most important to Logo (from its LISP ancestry) is *lists*.
- *Turtle graphics*. Logo has a turtle that lives "on-screen". This turtle responds to commands given to it. The turtle has the ability to leave behind a trail.

TI Logo Features

TI Logo adds the following features:

- *Sprites.* Sprites, like turtles, live on the screen. They do not leave behind a trail, but unlike ordinary turtles, can be given a color and shape and can move across the screen smoothly and continuously, without further program control.
- TI Logo II includes *music* commands with up to three voices plus a "drum". Music can be made to synchronize with graphics, and can be played concurrently with other actions without further program control.
- Integers. TI Logo represents numbers in the range of -32768 to 32767. All mathematical operations are integer operations.
- Music. TI Logo II adds the capability of music using three tone generators and a noise generator. Music can be synchronized with other processes or played concurrently.

Modes of Operation

TI Logo has three separate modes of operation. When you first enter TI Logo, you enter the immediate mode, and see the message:

WELCOME TO TI LOGO!

At this point you can type in immediate commands. To enter edit mode, for example to create a procedure called test, you enter "EDIT TEST" or "TO TEST". The screen will change from CYAN to GREEN with a red cursor. In the editor you may type as many procedures as you wish. Function-BACK returns to immediate mode.

The third mode is Turtle graphics mode. This is entered by typing "TELL TURTLE". The upper portion of the screen becomes a graphics tablet for the turtle. The lower six lines becomes a text window for you to type commands. The TI screen is 32 characters wide by 24 rows. Pixel resolution is 256 columns by 192 rows. To return to immediate mode, type "NOTURTLE".

TI Logo Primitives

*	Performs infix multiplication. See PRODUCT. Ex: PRINT 5 * 3 *15*
+	Performs infix addition. See SUM. Ex: PRINT 5 + 3 *8*
-	Performs infix subtraction. See DIFFERENCE. Ex: PRINT 5 - 2 *3*
.GC (Logo II)	Performs garbage collection to reclaim memory.
.HELP (Logo II)	Prints a list of all keywords in TI Logo
.NODES (Logo II)	Outputs number of currently free nodes.

/	Performs infix division. See QUOTIENT. Ex: PRINT 10 / 3 *3* (TI Logo only has integers in the range of -32768 to 32767)
:	Outputs value of its input. See THING. Ex: MAKE "APPLE 50 PRINT :APPLE *50*
;	Begins a comment.
<	Infix operator. Outputs true is its first input is less than its second. See LESS.
=	Infix operator. If both inputs are numbers, compares them to see if they are numerically equal. If both inputs are words, compares them to see if they are identical character strings. If both inputs are lists, compares them to see if their corresponding elements are equal. Outputs TRUE or FALSE. See IS.
>	Infix operator. Outputs true is its first input is greater than its second. See GREATER.
ALL	A list of sprites, used with TELL to direct all sprites (0-31). Ex: TELL :ALL CARRY :BALL Tells all sprites to look like the BALL shape.
BACK *number*, BK	Takes one number as input and moves active turtle or sprite back that many unites. Ex: BACK 100
BEEP	Starts computer playing a tone. See NOBEEP.
BIG (Logo II)	Magnifies all sprites to be twice their normal size. (32 x 32 pixels)
BOTH	Takes two inputs. Each input should be either TRUE or FALSE. Outputs TRUE if both are TRUE, otherwise outputs FALSE. Ex: PRINT BOTH (1 + 1 = 2)(5 = 4) *FALSE*
BUTFIRST *list*, BF	If input is a list, outputs a list containing all but the first element. If input is a word, outputs a word containing all but the first character. BF of empty list returns empty list. BF of single-character word returns empty list. Ex: PRINT BF [THIS IS A LIST] *IS A LIST* PRINT BF "ABRACADABRA *BRACADABRA*

BUTLAST *list*, BL	If input is a list, outputs a list containing all but the last element. If input is a word, outputs a word containing all but the last character. BL of empty list returns empty list. BL of single-character word returns empty list.
	Ex: PRINT BL [THIS IS A LIST] *THIS IS A* PRINT BL "ABRACADABRA *ABRACADABR*
BYE	Leaves TI Logo.
CALL value "word	Takes two inputs, the second which must be a word. Assigns the first input to be the value associated with the second input. See MAKE. Ex: CALL 7 "LUCKYNUMBER CALL [ALPHA BETA GAMMA] "TESTWORDS
CARRY number	Tells active sprite to "carry" the corresponding shape (a number from 0 through 25). See LOOKLIKE. Ex: CARRY :TRUCK
CHARNUM *"character*, CN	Takes a character as input and outputs the code number of that character. Ex: PRINT CN "A *65*
CHROMATIC (Logo II)	*(Default)* Changes meanings of pitch designations for music, making each unit a half-step. See MAJOR.
CLEARSCREEN, CS	Clears the screen.
COLORBACKGROUND *number*, CB	Sets the screen background to the color given (0-15).
COLOR	Outputs the color of the active sprite or foreground color of turtle. Does NOT return color of active tile.
CONTENTS (Logo II)	Outputs a list of all words in the workspace.
CONTINUE (Logo II)	Resumes execution after a pause is entered (from either function-AID or DEBUG).
DEBUG (Logo II)	Toggles debug state. If debug is off, an error returns control to top level. If debug is on, error returns control to current level for debugging purposes.
DEFINE *"word list*, DE	Takes two inputs, a name and a list. This word allows you to define a procedure programmatically. Usually the word TO or EDIT is used to define a procedure. Ex: DE "PTSUM [[:X :Y] [PRINT :X] [PRINT :X + :Y]] Defines the procedure: TO PTSUM :X :Y PRINT :X PRINT :Y + :Y

	PRINT :X + 1 END
DIFFERENCE	A prefix operation equivalent to -. Ex: PRINT DIFFERENCE 10 6 *4*
DOT *number number*	Takes two numeric inputs, x and y coordinates, and places a dot at the designated point on the turtle screen. Ex: DOT 30 30
DRUM *list* (Logo II)	Takes a list of numbers as input, and "signals" a "drumbeat" with the designated durations between beats. These are placed in the music buffer to be played by PLAYMUSIC. Ex: DRUM [3 4 6 8]
EACH *list*	Takes a list of commands as inputs, and runs the list for each active sprite. See YOURNUMBER. Ex: TELL :ALL EACH [SETHEADING 10 * YOURNUMBER]
EDIT *name*	Enters the procedure editor with a given procedure. If none specified, enters the editor with a blank screen. Ex: EDIT SQUARE
EITHER	Takes two inputs and outputs TRUE if at least one is true; outputs FALSE otherwise.
ELSE	Used in IF... THEN... ELSE. See IF.
END	Terminates a procedure definition typed into the editor.
ERASE *value*	Erases either a procedure or variable named. Ex: ERASE SQUARE erases procedure SQUARE ERASE :X erases variable X. ERASE "X erases both procedure X and variable X.
FALSE (Logo II)	Outputs word "FALSE For TI Logo: TO FALSE OUTPUT "FALSE END
FIRST *list*, F	Outputs the first element of a list or the first character of a word.
FORWARD *number*, FD	Moves the currently active turtle or sprite the number of units specified in the direction of its heading. Draws line if turtle's pen is down.
FPUT *value list*	Takes two inputs. The second must be a list. Outputs a list consisting of the first input followed by the elements of the second input. Basically inserts word or list1 into list2 at beginning of list2. Ex: PRINT FPUT [A B] [C D]

	Ex: PRINT [A B] [C D] [A B] C D Contrast with SENTENCE.
FREEZE	Stops motion of all sprites on the screen. See THAW.
GO *"word*	Transfers control within a procedure to the line designated by label given. Ex: TO TRIANGLE :STRING IF FIRST :STRING = :STRING THEN STOP LOOP: PRINT :STRING MAKE "STRING BUTFIRST :STRING GO "LOOP END
GREATER	Prefix form of >.
HEADING	Outputs heading of active sprite or turtle as a number between 0 and 360.
HIDETURTLE, HT	Makes the turtle pointer disappear.
HOME	Moves active turtle or sprite(s) to center of the screen. Additionally changes turtle heading to 0 (straight up).
IFF	Executes rest of line only if result of preceding TEST was false. See TEST.
IFT	Executes rest of line only if result of preceding TEST was true. See TEST.
IF	Used in the basic conditional form IF {condition} THEN {action1} ELSE {action2}. If {condition} is true then perform {action1} else perform {action2}. ELSE need not be present.
IS	Prefix form of =. Ex: IF IS 7 3 + 4 PRINT [YES]
JOY *number*	Takes one input number, specifying joystick 1 or 2. Outputs a number specifying joystick position. Using compass points: 0 - SW 1 - W 2 - NW 4 - S 5 - centered 6 - N 8 - SE 9 - E 10 - NE In Logo II, this number is increased by 16 to indicate the button on the joystick is pressed. **Warning:** Ensure ALPHA LOCK key is in up position.
LAST *list* LAST *"word*	Outputs last element of a list or the last character of a word
LEFT *number,* LT	Rotates active turtle or sprite *number* degrees counterclockwise.
LEGATO (Logo II)	Controls "dead time" inserted between notes. Note will sound for 1/60 second and 5/60 will be quiet. See STACCATO.
LENGTH *"word*	Outputs number of characters in a word or number of words

LENGTH *list* (Logo II)	in a list. TO LENGTH :X IF :X = [] OUTPUT 0 OUTPUT 1 + LENGTH BUTFIRST :X END
LESS	Prefix form of <.
LOOKLIKE *number*	See CARRY.
LOOPMUSIC (Logo II)	Plays music in the music buffer repeatedly. To stop music, use SETVOICE 0.
LPUT *value list*	Takes two inputs. The second must be a list. Outputs a list consisting of the second input followed by the elements of the first input. Basically inserts word or list1 into list2 at end of list2. Ex: PRINT LPUT [A B] [C D] C D [A B] Contrast with SENTENCE.
MAJOR (Logo II)	Changes meanings of the pitch designations. In MAJOR mode, 0 is middle C and each unit is a note on the C scale. See CHROMATIC.
MAKE *"word value*	First input is a word, second input is value to be associated with the word. See CALL.
MAKECHAR *number,* MC	Takes a *number* in the range of 0 - 255. Places you into the Character Editor, to change design of character *number.*
MAKESHAPE *number,* MS	Takes a *number* in the range of 0 - 25. Places you into the Shape Editor, to change design of sprite pattern *number*.
MUSIC *list1 list2* (Logo II)	Takes as input two lists. *List1* is a list of pitches. *List2* is a list of durations. These are placed into the Music Buffer. If one list is longer than the other, it is truncated. If a single number is given for *list2*, that duration is used for each pitch. Volume is taken as the value specified by the previous SETVOLUME command.
NOBEEP	Stops the tone started by BEEP.
NOTE *number1 number2 number3* (Logo II)	Takes three numbers as input, specifying duration, pitch, and volume for a note, and places it into the Music Buffer.
NOTURTLE	Exits turtle mode.
NOT	Outputs TRUE if its input is FALSE, FALSE if its input is TRUE. Ex: IF NOT (1=2) PRINT "HELP *HELP*
NUMBER?	Outputs TRUE if its input is a number, FALSE otherwise.
NUMBEROF *value*	Outputs number. Usually used with WHO. Ex: PRINT NUMBEROF WHO

OUTPUT *expr*, OP	Takes one input. Causes the current procedure to stop and output the result, *expr*, to the calling procedure.
PA	Prints all procedures and names.

PENDOWN, PD	Causes the turtle to leave a trail when it moves.
PENERASE, PE	Causes the turtle to erase any points it passes over.
PENREVERSE, PR	Causes the turtle to reverse any point it passes over.
PENUP, PU	Causes the turtle to move without leaving a trail.
PLAYMUSIC, PM (Logo II)	Plays the music in the Music Buffer.
PLAYNOTE (Logo II)	Plays one note from the Music Buffer, then waits for the duration of the note.
PN	Prints all currently defined names.
PO *name*	Takes a procedure name and prints the definition.
PP	Prints the title lines of all defined procedures.
PRINT *value*	Prints its input and moves cursor to the next screen line.
PRINTCHAR *number*, PC	Takes a tile number as input, and prints the corresponding character at the current cursor position.
PRINTOUT (Logo II)	Prints all procedures on a thermal printer or device.
PRODUCT	Prefix version of *.
PUTTILE *number row col,* PT	Takes a tile number row and column numbers. Places the tile at row and column.
QUOTIENT	Prefix version of /.
RANDOM	Outputs a number between 0 and 9.
RC?	Outputs TRUE if a keyboard character is pending (keyboard buffer not empty), else outputs FALSE.
READCHAR, RC	Outputs the next character in the keyboard buffer. If the buffer is empty, waits for a keypress.
READLINE, RL	Returns input line, output as a list.
RECALL	Reads information from cassette or disk. Procedure prompts you for: Recall 1. Procedures 2. Shapes 3. Both See SAVE.
REPEAT *number list*	Executes *list* for *number* iterations.
REST *number* (Logo II)	Inserts a rest of *number* duration in the Music Buffer.
REVERSE *"word* REVERSE *list* (Logo II)	Outputs the characters of a word in reverse order. Outputs the words of a list in reverse order. Ex: PRINT REVERSE "APPLESAUCE

	ECUASELPPA PRINT REVERSE [A [B C] [D E]] *[D E] [B C] A*
RIGHT *number,* RT	Rotates the active turtle or sprite *number* many degrees clockwise.
ROTATE *"word* ROTATE *list* (Logo II)	Outputs word with the first character moved to the end, i.e. WORD (BUTFIRST :X) (FIRST :X). Outputs list with the first item moved to the end, i.e. LPUT (FIRST :X)(BUTFIRST :X) Ex: PRINT ROTATE "APPLESAUCE *PPLESAUCEA* PRINT ROTATE [A [B C] [D E]] *[B C] [D E] A*
RUN *list*	Executes *list* as if it were typed at the command line.
SAVE	Stores information on cassette or disk. See RECALL.
SENTENCE *value value,* SE	Combines lists into a single list. Words are regarded as one-element lists. To combine a variable number of lists, enclose expression in parentheses. Ex: PRINT SENTENCE "HELLO "THERE *HELLO THERE* PRINT SENTENCE [THIS IS] [A LIST] *THIS IS A LIST* PRINT (SENTENCE "THIS [IS] [A LIST]) *THIS IS A LIST* PRINT SENTENCE [[HERE IS] A] [NESTED LIST] *[HERE IS] A NESTED LIST*
SETCOLOR *number* SETCOLOR [*number1 number2*], SC	*Number* is 0 through 15. For sprites, sprite assumes the color specified. For the turtle, *number1* is the pen color, *number2*, if present, is the background color of the tiles used for the pen. For tiles, *number1* is the foreground color and *number2* is the background color of the tile (actually for the set of eight characters the tile belongs to).
SETHEADING *number,* SH	Rotates the active sprite or turtle to point in the direction *number* degrees.
SETSPEED *number,* SS	Takes a *number* in the range of -128 through 127. Sets the speed of the active sprite.
SETTEMPO *number* (Logo II)	Sets the tempo in *number* of beats per minute. Default is 300.
SETVOICE *number* (Logo II)	Takes a *number* from 0 to 4. 0 clears the music buffer. 1-3 selects one of three voices. 4 selects the noise generator

(Logo II)	selects one of three voices. 4 selects the noise generator. Subsequent note commands will be directed to that voice.
SETVOLUME *number* (Logo II)	Takes a *number* from 0 to 15 to set the volume. 0 (default) is the softest, 15 is the loudest. As in SETVOICE, subsequent note commands will use the last SETVOLUME.
SHAPE	Returns the shape number of the active sprite.
SHOWTURTLE, ST	Makes the turtle pointer appear visible.
SIZE (Logo II)	Outputs 16 if sprites are SMALL and 32 if they are BIG.
SMALL (Logo II)	*Default.* Makes sprites 16 x 16 in size. See BIG.
SPEED	Outputs the speed of the currently active sprite.
SPRITE *number*	Takes a *number* in the range of 0 through 31. Used with TELL in order to direct commands to a sprite.
STACCATO (Logo II)	Controls "dead time" inserted between notes. In STACCATO, notes sound for 5/60 of a second and are silent for 1/60. Contract LEGATO.
STOP	Causes the current procedure to stop and return to the calling procedure.
SUM	Prefix version of +.
SV *number number*	Sets the x velocity and y velocity of the active sprite.
SXV *number*	Sets the x velocity of the active sprite.
SXY *number number*	Moves the active sprite or turtle to the specified point, where (0,0) is the center of the screen.
SX *number*	Moves the active sprite or turtle horizontally to the specified coordinate.
SY *number*	Moves the active sprite or turtle vertically to the specified coordinate
SYV *number*	Sets the y velocity of the active sprite.
TELL [*object*] *list*	Used to select an active object. Object, if present, specifies SPRITE, TILE, TURTLE, BACKGROUND. TELL used with a list of numbers directs actions to those sprites. Ex: TELL SPRITE 1 TELL TILE 50 TELL TURTLE TELL BACKGROUND TELL [1 5 8] TELL 10 TELL :ALL
TEST *expr*	Tests a condition to be used in conjunction with IFT and IFF.
TEXT *name*	Takes a procedure name as input and outputs the procedure text as a list. See DEFINE. Ex: TO PTSUM :X :Y PRINT :X PRINT :X + :Y

	END PRINT TEXT "PTSUM
	[:X :Y] [PRINT :X] [PRINT :X + :Y]
THAW	Restores motion that was stopped by FREEZE.
THEN	See IF.
THING? *Input*	Outputs TRUE if its input has a value associated with it.
THING *"word*	Outputs the value of its input. THING "XXX is the same as :XXX. See :.
TILE *number*	Used with TELL to designate an active tile. Ex: TELL TILE 100
TO *name*	Begins procedure definition. Enters edit mode.
TRACEBACK (Logo II)	When called within a procedure, prints the chain of procedure calls from the current procedure back to top level.
TRUE (Logo II)	Outputs the word "TRUE. For TI Logo: TO TRUE OUTPUT "TRUE END
TURTLE	Used with TELL in order to specify the turtle.
TYPE *list*	Like print, but does not move cursor to the next line after printing.
WAIT *number*	Waits *number* sixtieths of a second.
WHERE	If the turtle is active, outputs a list containing the x-coordinate, y-coordinate, and heading.
WHO	Outputs the active graphics object. Ex: TELL TURTLE PRINT WHO *TURTLE* TELL SPRITE 5 PRINT WHO *SPRITE 5* TELL [0 1 2] PRINT WHO [0 1 2] TELL TILE 100

	PRINT WHO *TILE 100*
WORD? *Input*	Outputs TRUE if *input* is a word. FALSE otherwise.
WORD *"word1 "word2*	Outputs a word that is the concatenation of *word1* and *word2.* Ex: PRINT "MISH "MASH *MISHMASH*
XCOR	Outputs x coordinate of the active turtle or sprite.
XVEL	Outputs x velocity of the active turtle or sprite.
YCOR	Outputs y coordinate of the active turtle or sprite.
YOURNUMBER, YN	Outputs the number of the currently active sprite. Normally used inside a command list with EACH.
YVEL	Outputs the y velocity of the active sprite.

The details on this page was provided by Ben Yates ti99_forever -at- yahoo.com, links and screen dumps were available Oct 2006 at

http://www.eskimo.com/~bjjly/logo.htm

The logo_sprites.htm doesn't seem to render under Firefox,
but it works under Konqueror... (hmm?)

Submenu

Sample Code

< TOP >