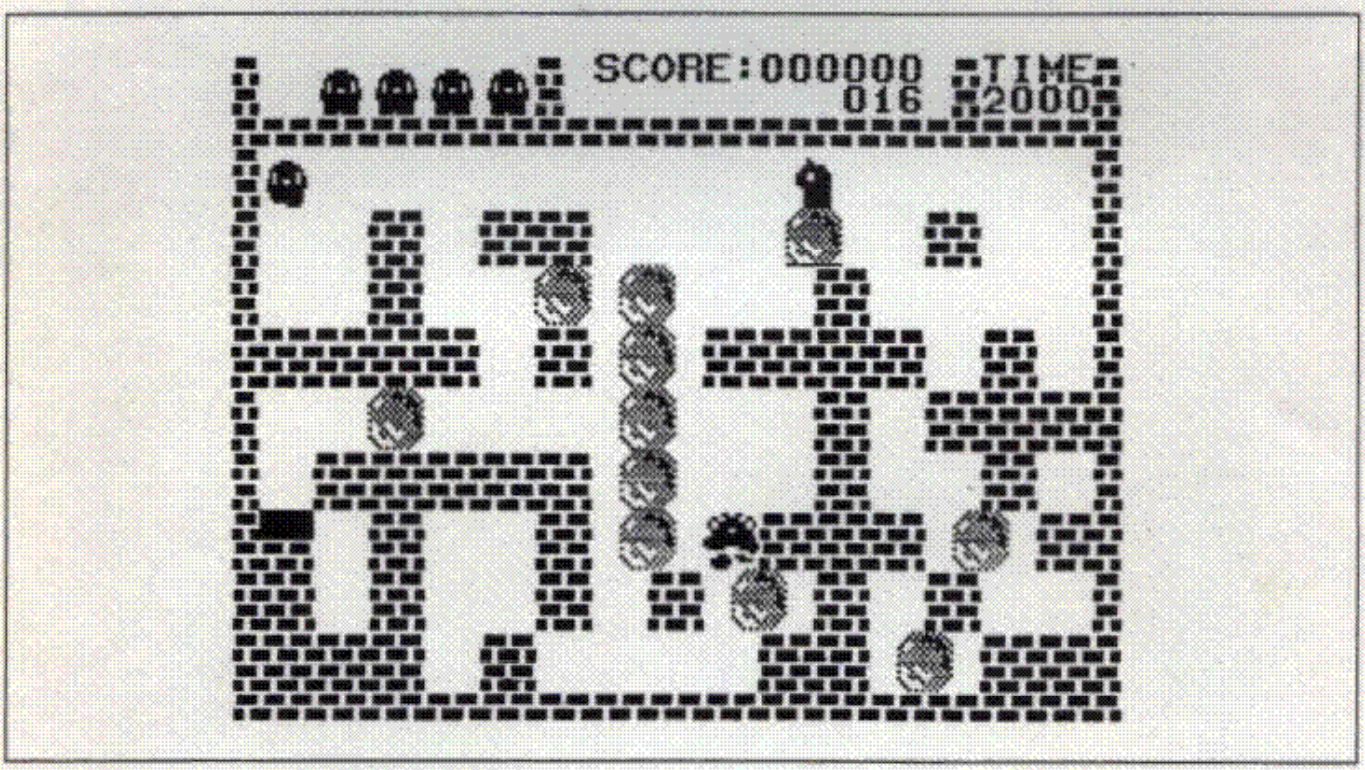
Journal of the ISTE Special Interest Group for Logo-Using Educators

LOGO EXCHANGE

September 1989 Volume 8 Number 1



International Society for Technology in Education

ISTE Publications

LOGO EXCHANGE

Volume 8 Number 1 Journal of the ISTE Special Interest Group for Logo-Using Educators September 1989

Logo Exchange is the journal of the International Society for Technology in Education Special Interest Group for Logo-using Educators (SIGLogo), published monthly September through May by ISTE, University of Oregon, 1787 Agate Street, Eugene, OR 97403-9905, USA.

POSTMASTER: Send address changes to Logo Exchange, UofO, 1787 Agate St., Eugene, OR 97403. Second-class postage paid at Eugene OR. USPS #000-554.

Contents

ISTE Membership (includes *The Computing Teacher*)

	U.S.	Non-U.S.
	28.50	36.00

SIGLogo Membership (includes *The Logo Exchange*)

	U.S.	Non-U.S.
ISTE Member Price	25.00	30.00
Non-ISTE Member Price	30.00	35.00

Send membership dues to ISTE. Add $2.50 for processing if payment does not accompany your dues. VISA and Mastercard accepted. Add $18.00 for airmail shipping.

Adding a Set of Alternative Conditional Primitives to LogoWriter

by Charles E. Crume and Cleborne D. Maddux

In the past, computer languages were written to satisfy the needs of computer programmers. However, as computer education has become more common, ideal languages for use in public schools must meet the needs of teachers and children. Therefore, characteristics such as speed, power, and compactness must be accompanied by simplicity and ease of learning. Logo is exemplary in this regard, since it was designed specifically with children in mind.

There are several dialects of Logo available, however, and later dialects, such as LogoWriter, have incorporated a number of changes to the original syntax. Teachers sometimes complain that favorite primitives from one dialect are not available in another.

The purpose of the this article is to demonstrate how common Logo primitives not included in LogoWriter, can be incorporated by creation of LogoWriter "tool procedures".

Tool Procedures

With most of the early computer languages such as BASIC, FORTRAN, and COBOL, adding primitives is difficult or impossible. In Logo, however, new primitives can be defined by simply writing a procedure to accomplish the desired task. An especially handy feature in LogoWriter is the ability to write tool procedures. Any LogoWriter page can be loaded into memory as a tool procedure. LogoWriter tool procedures are excellent for use with children since they do not appear on the flip side (where procedures are stored and edited). Consequently, they are not routinely accessible for editing (or accidental erasure). Then too, such tools remain in memory throughout a Logo session and take less work space than non-tool procedures.

Using tool procedures in LogoWriter is simple and straightforward. For example, if a LogoWriter page named MYTOOLS contained procedures named TEST, IFTRUE, and IFFALSE, the following command would be entered to load them as tool procedures:

```
GETTOOLS "MYTOOLS
```

As with any LogoWriter primitive, the GETTOOLS command can be entered from the keyboard or incorporated into any LogoWriter procedure. To verify that the procedures have been loaded properly as tool procedures, enter the command:

```
SHOW TOOLLIST
```

LogoWriter should respond:

```
[TEST IFTRUE IFFALSE]
```

TEST, IFTRUE, and IFFALSE Tool Procedures

Most Logo dialects include the TEST, IFTRUE, and IFFALSE primitives. LogoWriter, however, does not include these conditionals. Instead, only IF and IFELSE are provided. The primitive IF allows a list (enclosed in brackets) of actions to be executed only if the condition is true:

```
IF :A = 5
   [PRINT [YES, THE ANSWER IS 5.] ]
```

Some users may wish to designate one list of actions if the condition is true, and another list if it is false. In LogoWriter, the IFELSE primitive must be used:

```
IFELSE :A = 5
   [PRINT [YES, THE ANSWER IS 5.] ]
   [PRINT [SORRY, THAT IS NOT
    CORRECT.] ]
```

In the example above, if variable A equals 5, the actions in the first set of brackets will be executed. If A does not equal 5, the actions in the second set will be executed.

We have found that beginners are often confused by this LogoWriter syntax, and we have often wished that TEST, IFTRUE, and IFFALSE were available. We believe these latter primitives are superior because the logic is more sequential and straightforward, and permits less information to be placed on a single line. In dialects incorporating these primitives, the IFELSE example above would be written as follows:

```
TEST [:A = 5]
IFTRUE
   [PRINT [YES, THE ANSWER IS 5.] ]
IFFALSE
   [PRINT [SORRY, THAT IS NOT
    CORRECT.] ]
```

Therefore, we have written TEST, IFTRUE, and IFFALSE primitives for LogoWriter. These would be entered, named, and saved on a single LogoWriter page. To make these new primitives available, one uses the GETTOOLS command described above. The procedures follow:

```
TO TEST :MYLIST
IF NOT LIST? :MYLIST [(TYPE [TEST
   DOESN'T LIKE] :MYLIST [AS INPUT]
   CHAR 13) STOP]
IFELSE RUN :MYLIST
   [MAKE "TEST.RESULT "TRUE]
   [MAKE "TEST.RESULT "FALSE]
END

TO IFTRUE :LIST.OF.COMMANDS
IF NOT LIST? :LIST.OF.COMMANDS
   [(TYPE [IFTRUE DOESN'T LIKE]
   :LIST.OF.COMMANDS
   [AS INPUT] CHAR 13) STOP]
IF NAME? "TEST.RESULT
   [ IF :TEST.RESULT = "TRUE
   [ RUN :LIST.OF.COMMANDS ] ]
END

TO IFFALSE :LIST.OF.COMMANDS
IF NOT LIST? :LIST.OF.COMMANDS
   [(TYPE [IFFALSE DOESN'T LIKE]
   :LIST.OF.COMMANDS [AS INPUT] CHAR
   13) STOP]
IF NAME? "TEST.RESULT
   [ IF :TEST.RESULT = "FALSE
   [ RUN :LIST.OF.COMMANDS ] ]
END
```

All three of the above procedures require a single input that must be a list (enclosed in brackets). TEST is error trapped to insure that its input (MYLIST) is a list. If not, the error message "TEST DOESN'T LIKE (input) AS INPUT" will be displayed in the Command Center at the bottom of the screen. If the input is a list, TEST evaluates the list (via the RUN primitive) and assigns either TRUE or FALSE to the variable TEST.RESULT.

IFTRUE and IFFALSE are also error trapped to return an error message if their input is not a list. Both of these procedures use the NAME? primitive to verify that the variable TEST.RESULT exists. If it does exist, TEST.RESULT is checked to see if it has been assigned TRUE or FALSE. IFTRUE then runs the list of commands if TEST.RESULT is true. IFFALSE performs similarly if TEST.RESULT is FALSE.

The AND, OR, and NOT primitives as well as internal parentheses may be included in the input to TEST so long as all syntax conforms to LogoWriter rules. For example:

```
TEST [AND (:A = 5) (:B > 3)]
IFTRUE [PRINT "YES]
IFFALSE [PRINT "NO]
```

The parentheses following AND above are optional, just as they would be in any LogoWriter line containing AND. In the example above, if the A variable is equal to 5 and if the B variable is greater than 3, YES will be displayed. Otherwise, NO will be displayed.

Conclusions

We hope the above explanation and sample tool procedures will be helpful to teachers using LogoWriter in their classrooms. We are planning additional LogoWriter tool kits that we believe will simplify the understanding and learning of Logo.

Charles E. Crume, B.S.
Technical Consultant
University of Nevada System Computing Services

Cleborne D. Maddux, Ph.D.
Professor and Chairman
Department of Curriculum and Instruction
University of Nevada Reno