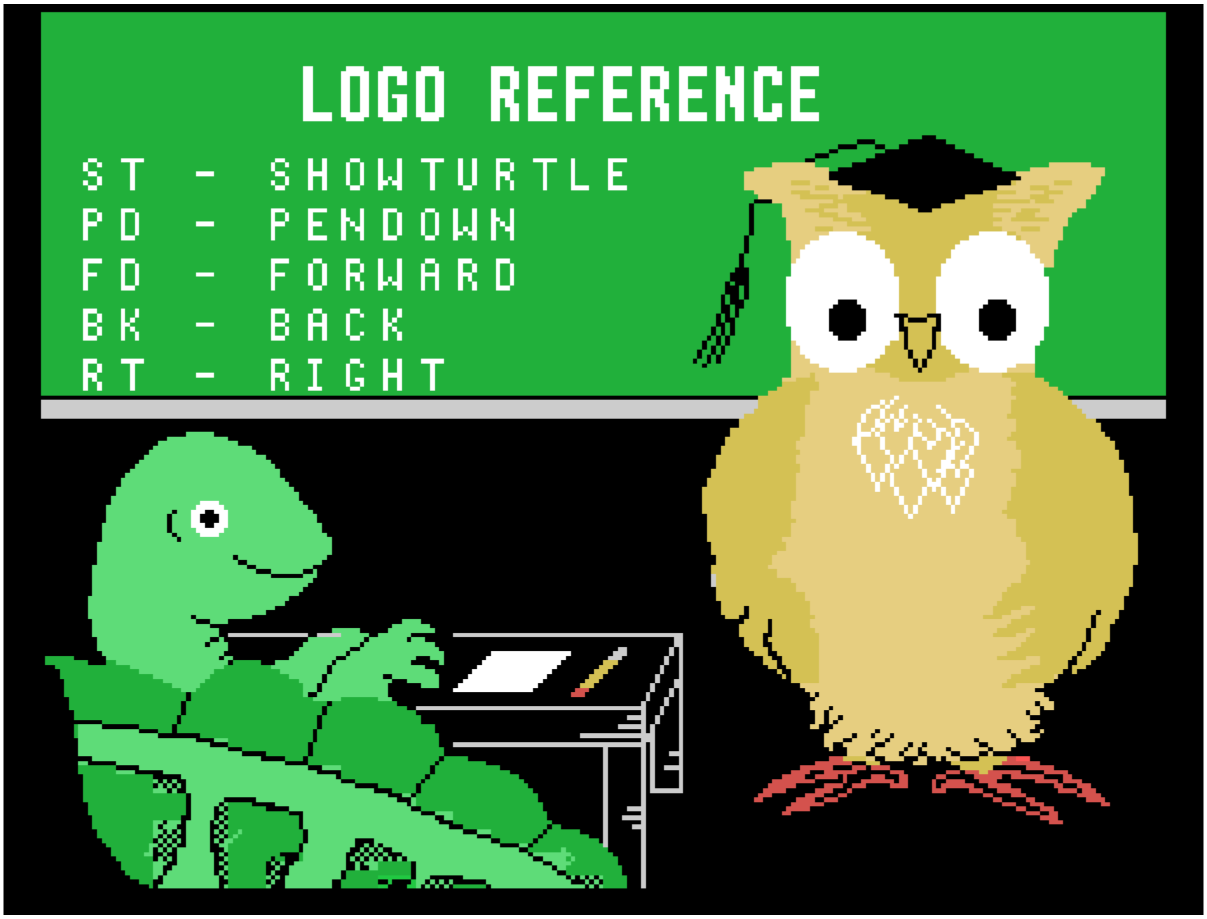
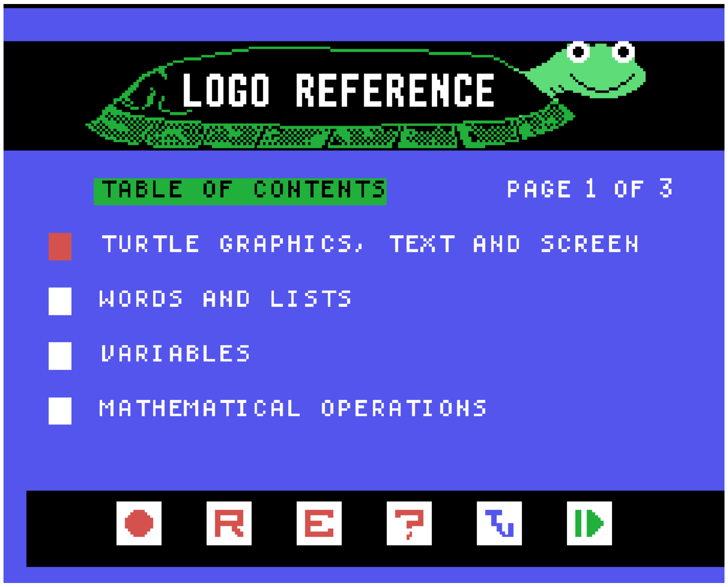
NABU

LOGO REFERENCE



LOGO REFERENCE

ST - SHOWTURTLE
PD - PENDOWN
FD - FORWARD
BK - BACK
RT - RIGHT



LOGO REFERENCE

TABLE OF CONTENTS PAGE 1 OF 3

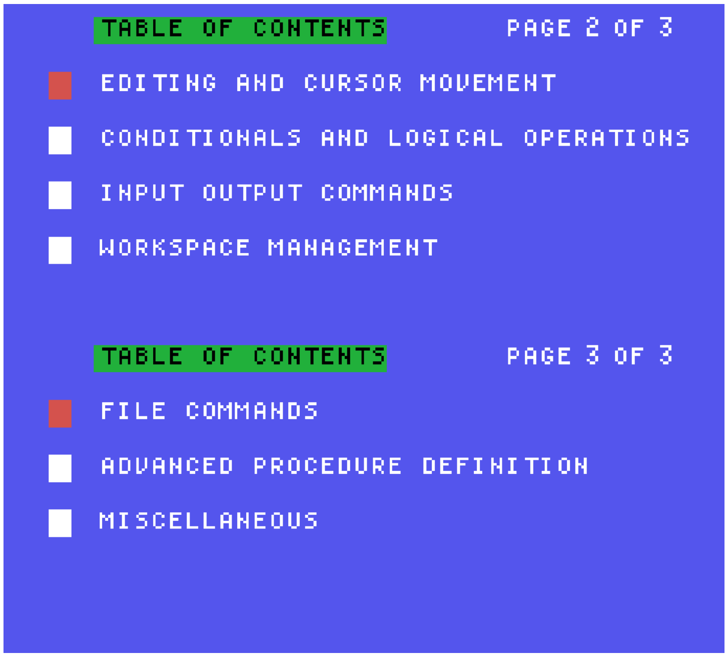


TABLE OF CONTENTS PAGE 2 OF 3

TABLE OF CONTENTS PAGE 3 OF 3

LOGO REFERENCE

READY TO GO? PRESS GO
NEED HELP? PRESS HELP
NEW CHOICE? PRESS EXIT
START AGAIN? PRESS RESTART

TABLE OF CONTENTS

PAGE 1 OF 3

TABLE OF CONTENTS

PAGE 2 OF 3

TABLE OF CONTENTS

PAGE 3 OF 3

LOGO REFERENCE HELP PAGE 1 OF 10

Use the joystick or the arrow keys, to
Move the red square over your choice,
Then press the fire button or press go.

- previous page or no previous
- restart, go to previous menu
- exit to nabu category menu
- help
- switch to television
- next page or no next

LOGO REFERENCE HELP PAGE 2 OF 10

The Logo Reference explains logo primitives (instructions) and how to use them. It also explains how to use the Sym and Ctrl keys to move around the screen and edit your work.

Information is divided into categories. Choose a category from the table of contents menu. Use the joystick or the arrow keys to move the red square to the category you want to view.

To display your selected category, press the fire button or the go key.

LOGO REFERENCE HELP PAGE 3 OF 10

The primitives are divided into categories as they appear in the Nabu Logo learner's guide. Each category contains an alphabetic list of primitives with brief explanations.

The ctrl and sym keys are listed alphabetically in the category called Editing and cursor movement

LOGO REFERENCE HELP PAGE 4 OF 10

A primitive can be:

- a **command**, which the computer recognizes as an instruction to do something.

Showturtle is an example of a **command**.

- an **operation**, which the computer recognizes as an instruction to do something and then produce a value.

Sum is an example of an **operation**.

LOGO REFERENCE HELP PAGE 5 OF 10

For every primitive, instructions are

shown in black on a green background. This is the information that you must type to tell the computer what to do. The name of the primitive is shown in upper case. If there is a short form, it is shown in () parentheses, for example:

showturtle (**st**)
You can type either **showturtle** or **st**.

LOGO REFERENCE HELP PAGE 6 OF 10

The computer sometimes needs more information than just the primitive before it can carry out your instructions.
Additional information is shown in lower case, for example:

Forward (fd) distance

You will type **forward** or **fd** plus the distance, which is the number of steps that you want the turtle to take.

LOGO REFERENCE HELP PAGE 7 OF 10

When the information that the computer needs is shown in the instruction as Object:

You can enter either a word, a number (for calculation) or a list.

Use double quotes before a word:

print "square

Use [ ] square brackets with a list:

print [31 turtles swam 62 miles]

LOGO REFERENCE HELP PAGE 8 OF 10

Use quotes or square brackets when a number is to be treated as a word:

print "49+1

print [49 + 1]

Both these tell the computer to print 49+1, without calculating the result. If you do not use quotes or brackets, the computer calculates the numbers:

print 49 + 1

The computer will print 50.

LOGO REFERENCE HELP PAGE 9 OF 10

Some primitives can accept several pieces of information. For example:

Sum number1 number2 ...

If you type **sum** 3 4, the computer outputs 7.

Notice the 3 dots. If you want to use more pieces of information, you must enclose them in parentheses.

For example, if you want the sum of 3, 4, 5 and 6 you type: **sum** (3 4 5 6)

LOGO REFERENCE HELP PAGE 10 OF 10

When you are ready to leave the current category and return to the table of contents to select another category, move the red square over <R> (restart) and press the fire button or the go key.

You can use the nabu keys for restart, help or exit. You can use the page keys to turn pages.

Leave help: red square to <E> (exit), press the fire button or the go key.

TURTLE GRAPHICS, TEXT AND SCREEN

ALL
Outputs the numbers of all sprites.

ASK spritenumber(s) instructionlist
Runs the commands in **instructionlist** temporarily making the current sprite **spritenumber(s)**.

BACK (BK) distance
Moves the current sprite **distance** steps back without changing its heading.

BG
Outputs a number representing the graphics background color.

BRUSH distance interval
Moves the current sprite forward a total of **distance** steps and stamps the shape of the current sprite at **interval** steps.

CHANGE.COLOR fromcolor tocolor
Changes everything on the graphics screen, except sprites, that is **fromcolor** to **tocolor**.

CLEAN
Erases everything on the graphics screen without affecting the sprite(s).

CLEARSCREEN (CS)
Erases everything on the graphics screen and returns the current sprite(s) to the home position [0 0], sets the heading to 0 and the speed to 0.

CLEARTEXT
Erases all text on the textmode screen and on the text portion of the splitscreen.

COLOR
Outputs a number representing the color of the current sprite.

COLOR.OVER
Outputs the number of the color of the part of the graphics screen that the current sprite is over.

COPYSH fromshapenumber toshapenumber
Copies the shape of **fromshapenumber** to **toshapenumber**.

CURSOR
Outputs the position of the cursor in the order X, Y.

DISTANCE position
Outputs the **distance** from position to the current sprite.

DOT position
Puts a dot at **position**.

EACH instructionlist
Causes all current sprites to sequentially run **instructionlist**.

FILL
Fills the enclosed region containing the current sprite with the color of the sprite's pen.

FORWARD (FD) distance
Moves the current sprite(s) forward **distance** steps without changing heading.

FREEZE
Stops all moving sprites.

GRMODE
Returns the screen from textmode to the previous graphics screen (mixed or split).

HEADING
Outputs the heading of the current sprite.

HIDETURTLE (HT)
Makes the current sprite(s) invisible.

HOME
Returns the current sprite(s) to the home position [0 0], setting the heading to 0 and speed to 0.

LEFT (LT) degrees
Turns the current sprite(s) left through **degrees**.

MIX
Devotes the entire screen to graphics.

MOVE time speed
Starts the current sprite(s) moving forward at **speed** and stops the movement after **time**.

PEN
Outputs the state of the pen of the current sprite.

PENCOLOR (PC)
Outputs the number of the current pen color.

PENDOWN (PD)
Puts the pen down so that it may draw.

PENERASE (PE)
Puts the eraser down so that it will erase any drawing it contacts.

PENUP (PU)
Lifts the pen so that it cannot draw.

POS
Outputs the position of the current sprite.

RIGHT (RT) degrees
Turns the current sprite(s) left through **degrees**.

SETBG color
Sets the background **color** of the graphics screen to color.

SETCOLOR (SETC) color
Sets the color of the current sprite(s) **color**.

SETCURSOR position
Moves the cursor to **position**.

SETHEADING (SETH) degrees
Sets the heading of the current sprite **degrees**.

SETPC color
Sets the pen to **color**.

SETPOS position
Puts the current sprite(s) at the coordinates **position**.

SETSHAPE (SETSH) shapenumber
Sets the shape of the current sprite(s) **shapenumber** and causes the sprite(s) to be displayed.

SETSPEED (SETS) speed
Starts the current sprite(s) moving forward at **speed**, or back if speed is a negative number, or stops movement if speed is 0.

SETTC color(s)
Sets the text color to **color**. If a list of two colors, the first sets the text color and the second sets the "paper" color.

SETX X coordinate
Sets the position of the current sprite **X coordinate** without changing the Y coordinate.

SETXVEL speed
Sets the horizontal component of the velocity of the current sprite(s) to **speed**. May result in a heading change.

SETY Y coordinate
Sets the position of the current sprite **Y coordinate** without changing the X coordinate.

SETYVEL speed
Sets the vertical component of the velocity of the current sprite(s) to **speed**. May result in a heading change.

SHADE
Fills an enclosed region containing the current sprite with the shape and the color of the current sprite.

SHAPE
Outputs the shape number of the current sprite.

SHOW
Gives the current sprite the shape it had with the most recent use of SETSHAPE.

SHOWNP
Outputs TRUE if the current sprite is showing on the screen. Otherwise FALSE.

SHOWTURTLE (ST)
Causes the current sprite(s) to appear with the turtle shape.

SLIDE distance time
Moves the current sprite(s) forward **distance** steps over a period equal to **time**.

SNAP
Causes the current sprite's shapenumber to assume the shape of whatever drawing the sprite is over.

SPEED
Outputs the speed of the current sprite.

SPLITSCREEN
Restricts the bottom 4 lines of the screen in graphics mode exclusively to text.

STAMP
Leaves an image on the screen of the current sprite in the current pen color.

TELL spritenumber(s)
Makes **spritenumber(s)** the current sprite(s).

TEXTCOLOR (TC)
Outputs the number of the color of the text.

TEXTMODE
Changes the screen from graphics mode to text mode, devoting the entire screen to text.

THAW
Restores motion to sprites frozen with FREEZE.

TOUCHINGP spritenumber
Outputs TRUE if the grids of **spritenumber** and the current sprite are overlapping.

TOWARDS position
Outputs the heading required for the current sprite to point toward **position**.

WHO
Outputs the number(s) of the current sprite(s).

XCOR
Outputs the X coordinate of the current sprite's position.

XVEL
Outputs the horizontal component of the speed of the current sprite.

YCOR
Outputs the Y coordinate of the current sprite's position.

YVEL
Outputs the vertical component of the speed of the current sprite.

WORDS AND LISTS

ASCII character
Outputs the ASCII code for **character**.

BUTFIRST (BF) object
Outputs all but the first element of **object**.

BUTLAST (BL) object
Outputs all but the last element of **object**.

CHAR asciicode
Outputs the character whose ASCII code is **asciicode**.

COUNT list
Outputs the number of elements in **list**.

EMPTYP object
Outputs TRUE if **object** is an empty list or empty word.

EQUALP object1 object2
Outputs TRUE if **object1** and **object2** are equal.

FIRST object
Outputs the first element of **object**.

FPUT object list
Outputs a new list formed from putting **object** in front of **list**.

ITEM nth list
Outputs the element in **nth** position of **list**.

LAST object
Outputs the last element of **object**.

LIST object1 object2 ...
Outputs a list of its inputs.

LISTP object
Outputs TRUE if **object** is a list.

LPUT object list
Outputs new list formed with
object placed at end of list.

MEMBERP object list
Outputs TRUE if **object** is an element of **list**.

NAME object name
Puts object in the variable
name (opposite order of MAKE).

NUMBERP object
Outputs TRUE if **object** is a number.

SENTENCE (SE) object1 object2 ...
Outputs a list of its inputs.

WORD word1 word2 ...
Outputs a word made up of its inputs.

WORDP object
Outputs TRUE if **object** is a word.

VARIABLES

LOCAL name ...
Makes variable **name** local to a
procedure.

MAKE name object
Makes a global variable **name**
with a value of object.

NAME object name
Puts **object** in the variable named **name**.

NAMEP name
Outputs TRUE if **name** has a value.

THING name
Outputs object referred to by name.

MATHEMATICAL OPERATIONS

ARCTAN number
Outputs the arctangent of **number**.

COS degrees
Outputs the cosine of **degrees**.

INT number
Outputs the integer portion of **number**.

PRODUCT number1 number2 ...
Outputs the product of its inputs.

QUOTIENT number1 number2
Outputs the integer portion of
number1 divided by **number2**.

RANDOM number
Outputs any positive integer
(including 0) less than **number**.

REMAINDER number1 number2
Outputs the remainder of **number1** divided by **number2**.

RERANDOM
Makes RANDOM behave reproducibly.

ROUND number
Outputs **number** rounded off to the nearest integer.

SIN degrees
Outputs the sine of **degrees**.

SQRT number
Outputs the square root of **number**.

SUM number1 number2 ...
Outputs the sum of its inputs.
a + b
Outputs a plus b.
a - b
Outputs a minus b.
a * b
Outputs a times b.
a / b
Outputs a divided by b.
a < b
Outputs TRUE if a is less than b.
object1 = object2
Outputs TRUE if **object1** equals **object2**.
a > b
Outputs TRUE if a is greater than b.

EDITING AND CURSOR MOVEMENT

ED
Starts Logo text editor with same display as when editor last used.

ED name(s)
Starts Logo text editor containing the named procedures.

EDITCHAR (EC) character
Starts Logo character editor containing **character**.

EDITSHAPE (ES) shapenumber
Starts Logo shape editor containing **shapenumber**.

EDNS
Starts Logo variable editor containing all defined variables.

EDNS package(s)
Starts Logo variable editors containing variables in **package(s)**.

END
Tells Logo you are at the end of a procedure definition.

TO procedurename
Begins the definition of a procedure named **procedurename**.

CURSOR MOVEMENT

CTRL-A
Moves cursor to the beginning of the current line.

CTRL-B
Moves cursor back one space.

CTRL-C
Exits from editor reading the buffer as if typed in.

CTRL-D
Erases character at the cursor position.

CTRL-E
Moves cursor to the end of the current line.

CTRL-F
Moves cursor forward one space.

CTRL-K
Erases everything on the current line
to the right of the cursor.

CTRL-N
Moves cursor down to next line in editor.

CTRL-O
Opens a new line at the position of the cursor in editor.

CTRL-P
Moves cursor up to the previous line in editor.

CTRL-Q
Quotes next character you type,printing a backslash before echoing the character.

CTRL-V
Scrolls the screen to the next page in the editor.

SYM-<
Moves the cursor to the beginning of the first procedure in editor.

SYM->
Moves the cursor to the end of the last procedure in editor.

UP ARROW
Moves the cursor to the previous line in editor.

DOWN ARROW
Moves the cursor to the next line in editor.

LEFT ARROW
Moves the cursor back one space.

RIGHT ARROW
Moves the cursor to the right one space.

PAGE-RIGHT key
Moves the cursor to the end of the current line.

PAGE-LEFT key
Moves the cursor to the beginning of the current line.

DELETE key
Erases the character to the left of the cursor.

NO key
Empties the filled-in square under the
cursor in the character and shape
editors.

YES key
Fills in the square under the cursor
in the character and shape editors.

CONDITIONALS AND LOGICAL OPERATIONS

CO
Continues a procedure after a pause.

CO object
Resumes a procedure after a pause with **object** being the output from PAUSE.

CTRL-G
Stops the execution of a procedure and returns control to top level.

CTRL-Z
Temporarily stops the execution of a running procedure.

IF condition list1 (list2)
If condition is TRUE, run **list1** otherwise run **list2**

IFFALSE (IFF) list
Runs **list** if most recent TEST was FALSE.

IFTRUE (IFT) list
Runs **list** if most recent TEST was TRUE.

OUTPUT (OP) object
Returns control to the caller with **object** as the output.

PAUSE
Makes a procedure pause.

REPEAT number list
Runs **list number** of times.

RUN list
Runs **list;** outputs what **list** outputs.

STOP
Stops execution of a procedure and returns control to the caller.

TEST condition
Remembers whether **condition** is TRUE or FALSE.

LOGICAL OPERATIONS

AND condition1 condition2 ...
Outputs TRUE if all its inputs are TRUE.

NOT condition
Outputs TRUE if **condition** is FALSE.

OR condition1 condition2 ...
Outputs TRUE if any of its inputs are TRUE.

INPUT OUTPUT COMMANDS

CHARP
Outputs TRUE if a key has been pressed but not yet read.

FIREP number
Outputs TRUE if fire button on games controller **number** is pressed.

FPRINT object
object is a list, displays **object** with enclosing square brackets; otherwise acts like PRINT.

JOY number
Outputs the value representing the position of the joystick on games controller **number**.

PRINT (PR) object ...
Displays **object** and puts cursor at a new line (removes brackets from around list objects).

READCHAR (RC)
Outputs a character typed by user, waiting if necessary.

READLIST (RL)
Outputs line typed by user, waiting if necessary.

STARTPRINT
Transfers output to the printer (printer must be connected and selected).

STOPPRINT
Stops output to the printer.

TYPE object1 ...
Displays inputs as for PRINT but does not start new line.

WAIT time
Pauses for period equal to **time**.

WORKSPACE MANAGEMENT

BURY package
Buries all the procedures contained in **package**.

ERALL
Erases all procedures not buried.

ERALL package(list)
Erases all procedures in **package(list)**.

ER procedurename(s)
Erases named procedures.

ERN variablename(s)
Erases named variables.

ERNS
Erases all unburied variables.

ERNS package(list)
Erases variables in named packages.

ERPS
Erases all unburied procedures.

ERPS package(s)
Erases all procedures in named packages.

PACKAGE packagename procedurename(s)
Puts **procedurename(s)** in **packagename**.

PKGALL package
Puts in **package** everything that is not already packaged.

PO name(s)
Prints the definitions of the named procedures.

POALL
Prints the definition of all unburied procedures and the values of all unburied variables.

POALL package(s)
Prints the definitions of all procedures and the names of all variables in **packages(s)**.

PONS
Prints the names and values of all unburied variables.

PONS package(s)
Prints the names and values of all variables in **package(s)**.

POPS
Prints the definitions of all unburied procedures.

POPS package(s)
Prints the definitions of all procedures in **package(s)**.

POTS
Prints the title lines of all unburied procedures.

POTS package(s)
Prints the title lines of all procedures **package(s)**.

UNBURY
Unburies all the procedures in the workspace.

UNBURY package
Unburies the procedures in **package**.

FILE COMMANDS

CATALOG
Displays the names of all files on a diskette in the current disk drive.

DISK
Outputs a number identifying which disk drive or Cable is active.

ERASEFILE (ERF) filename.extension
Erases the text (.LOG), character, shape or picture file named **filename.extension** from the diskette.

LOAD filename
Loads the .LOG file named **filename** from diskette back into the packages they came from.

LOAD filename packagename
Loads the .LOG file named **filename** from the diskette into the package named **packagename**.

LOADCHAR filename
Loads the character file named **filename** from diskette.

LOADSH filename
Loads the shape file named **filename** from diskette.

LOADPICT filename
Loads the picture file named **filename** from diskette.

SAVE filename
Writes the whole workspace to a .LOG

file **filename** on diskette.

SAVE filename package(s)
Writes **package(s)** to **filename** on diskette.

SAVECHAR filename
Writes the current character set to a .CHR file named **filename** on diskette.

SAVESH filename
Writes the current set of shapes to a .SHA file named **filename** on diskette.

SAVESH filename shapelist
Writes the listed shapes in **shapelist** into a .SHA file named filename.

SAVEPICT filename
Writes the current graphics display to a .PIC file named **filename** on a diskette.

SETDISK drivenumber
Sets the current drive to **drivenumber** which can represent Cable or floppy disk drives.

ADVANCED PROCEDURE DEFINITION

COPYDEF newname name
Copies the definition of **name** onto **newname**.

DEFINE name list
Makes **list** the definition of **name**.

PRIMITIVEP name
Outputs TRUE if **name** is a primitive.

TEXT name
Outputs definition of procedure **name** as a list.

MISCELLANEOUS

DEFINEDP word
Outputs TRUE if **word** is the name of a procedure.

PROCEDUREP name
Outputs TRUE if **name** is the name of a procedure.

GO word
Transfers control to LABEL **word**.

LABEL word
Labels a line for use by GO.

NODES
Outputs the number of free nodes.

RECYCLE
Performs a "garbage collection", freeing up as many nodes as possible.

REPARSE
Reparse causes Logo to reread and reinterpret all procedures in the workspace immediately.

.CONTENTS
Outputs a list of all the variables, procedures, primitives and general contents of memory.

.VEXAMINE N
Outputs the contents of a video display processor address **N**.

.VDEPOSIT N byte
Writes **byte** into the video display processor address **N**.

KEYWORDS LIST

TURTLE GRAPHICS, TEXT AND SCREEN

ALL
ASK spritenumber(s) instructionlist
BACK (BK) distance
BG
BRUSH distance interval
CHANGE.COLOR fromcolor tocolor
CLEAN
CLEARSCREEN (CS)
CLEARTEXT
COLOR
COLOR.OVER
COPYSH fromshapenumber toshapenumber
CURSOR
DISTANCE position
DOT position
EACH instructionlist
FILL
FORWARD (FD) distance
FREEZE
GRMODE
HEADING
HIDETURTLE (HT)
HOME
LEFT (LT) degrees
MIX
MOVE time speed
PEN
PENCOLOR (PC)
PENDOWN (PD)
PENERASE (PE)
PENUP (PU)
POS
RIGHT (RT) degrees
SETBG color
SETCOLOR (SETC) color
SETCURSOR position
SETHEADING (SETH) degrees
SETPC color
SETPOS position
SETSHAPE (SETSH) shapenumber
SETSPEED (SETS) speed
SETTC color(s)
SETX X coordinate
SETXVEL speed
SETY Y coordinate
SETYVEL speed
SHADE
SHAPE
SHOW
SHOWNP
SHOWTURTLE (ST)
SLIDE distance time
SNAP
SPEED
SPLITSCREEN
STAMP
TELL spritenumber(s)
TEXTCOLOR (TC)
TEXTMODE
THAW
TOUCHINGP spritenumber
TOWARDS position
WHO
XCOR
XVEL
YCOR
YVEL

WORDS AND LISTS

ASCII character
BUTFIRST (BF) object
BUTLAST (BL) object
CHAR asciicode
COUNT list
EMPTYP object
EQUALP object1 object2
FIRST object
FPUT object list
ITEM nth list
LAST object
LIST object1 object2 ...
LISTP object
LPUT object list
MEMBERP object list
NAME object name
NUMBERP object
SENTENCE (SE) object1 object2 ...
WORD word1 word2 ...
WORDP object

VARIABLES

LOCAL name ...
MAKE name object
NAME object name
NAMEP name
THING name

MATHEMATICAL OPERATIONS

ARCTAN number
COS degrees
INT number
PRODUCT number1 number2 ...
QUOTIENT number1 number2
RANDOM number
REMAINDER number1 number2
RERANDOM
ROUND number
SIN degrees
SQRT number
SUM number1 number2 ...
a + b
a - b
a * b
a / b
a < b
object1 = object2
a > b

EDITING AND CURSOR MOVEMENT

ED
ED name(s)
EDITCHAR (EC) character
EDITSHAPE (ES) shapenumber
EDNS
EDNS package(s)
END
TO procedurename

CURSOR MOVEMENT

CTRL-A
CTRL-B
CTRL-C
CTRL-D
CTRL-E
CTRL-F
CTRL-K
CTRL-N
CTRL-O
CTRL-P
CTRL-Q
CTRL-V
SYM-<
SYM->
UP ARROW
DOWN ARROW
LEFT ARROW
RIGHT ARROW
PAGE-RIGHT key
PAGE-LEFT key
DELETE key
NO key
YES key

CONDITIONALS AND LOGICAL OPERATIONS

CO
CO object
CTRL-G
CTRL-Z
IF condition list1 (list2)
IFFALSE (IFF) list
IFTRUE (IFT) list
OUTPUT (OP) object
PAUSE
REPEAT number list
RUN list
STOP
TEST condition

LOGICAL OPERATIONS

AND condition1 condition2 ...
NOT condition
OR condition1 condition2 ...

INPUT OUTPUT COMMANDS

CHARP
FIREP number
FPRINT object
JOY number

PRINT (PR) object ...
READCHAR (RC)
READLIST (RL)
STARTPRINT
STOPPRINT
TYPE object1 ...
WAIT time

WORKSPACE MANAGEMENT

BURY package
ERALL
ERALL package(list)
ER procedurename(s)
ERN variablename(s)
ERNS
ERNS package(list)
ERPS
ERPS package(s)
PACKAGE packagename procedurename(s)
PKGALL package
PO name(s)
POALL
POALL package(s)
PONS
PONS package(s)
POPS
POPS package(s)
POTS
POTS package(s)
UNBURY
UNBURY package

FILE COMMANDS

CATALOG
DISK
ERASEFILE (ERF) filename.extension
LOAD filename
LOAD filename packagename
LOADCHAR filename
LOADSH filename
LOADPICT filename
SAVE filename
SAVE filename package(s)
SAVECHAR filename
SAVESH filename
SAVESH filename shapelist
SAVEPICT filename
SETDISK drivenumber

ADVANCED PROCEDURE DEFINITION

COPYDEF newname name
DEFINE name list
PRIMITIVEP name
TEXT name

MISCELLANEOUS

DEFINEDP word
PROCEDUREP name
GO word
LABEL word
NODES
RECYCLE
REPARSE
.CONTENTS
.VEXAMINE object
.VDEPOSIT N a

LOGO KEYWORDS LIST

TURTLE GRAPHICS, TEXT AND SCREEN ALL ASK spritenumber(s) instructionlist BACK (BK) distance BG BRUSH distance interval CHANGE.COLOR fromcolor tocolor CLEAN CLEARSCREEN (CS) CLEARTEXT COLOR COLOR.OVER COPYSH fromshapenumber toshapenumber CURSOR DISTANCE position DOT position EACH instructionlist FILL FORWARD (FD) distance FREEZE GRMODE HEADING HIDETURTLE (HT) HOME LEFT (LT) degrees MIX MOVE time speed PEN PENCOLOR (PC) PENDOWN (PD) PENERASE (PE) PENUP (PU) POS RIGHT (RT) degrees SETBG color SETCOLOR (SETC) color SETCURSOR position SETHEADING (SETH) degrees SETPC color SETPOS position SETSHAPE (SETSH) shapenumber SETSPEED (SETS) speed SETTC color(s) SETX xcoordinate SETXVEL speed SETY ycoordinate SETYVEL speed SHADE SHAPE SHOW SHOWNP SHOWTURTLE (ST) SLIDE distance time SNAP SPEED SPLITSCREEN STAMP TELL spritenumber(s) TEXTCOLOR (TC) TEXTMODE THAW TOUCHINGP spritenumber TOWARDS position WHO XCOR XVEL YCOR YVEL

WORDS AND LISTS ASCII character BUTFIRST (BF) object BUTLAST (BL) object CHAR asciicode COUNT list EMPTYP object EQUALP object1 object2 FIRST object FPUT object list ITEM nth list LAST object LIST object1 object2 ... LISTP object LPUT object list MEMBERP object list NAME object name NUMBERP object SENTENCE (SE) object1 object2 ... WORD word1 word2 ... WORDP object

VARIABLES LOCAL name ... MAKE name object NAME object name NAMEP name THING name

MATHEMATICAL OPERATIONS ARCTAN number COS degrees INT number PRODUCT number1 number2 ... QUOTIENT number1 number2 RANDOM number REMAINDER number1 number2 RERANDOM ROUND number SIN degrees SQRT number SUM number1 number2 ... a + b a **-** b a * b a / b a < b object1 = object2 a > b

EDITING AND CURSOR MOVEMENT ED ED name(s) EDITCHAR (EC) character EDITSHAPE (ES) shapenumber EDNS EDNS package(s) END TO procedurename

CURSOR MOVEMENT CTRL-A CTRL-B CTRL-C CTRL-D CTRL-E CTRL-F CTRL-K CTRL-N CTRL-O CTRL-P CTRL-Q CTRL-V SYM-< SYM-> UP ARROW DOWN ARROW LEFT ARROW RIGHT ARROW PAGE-RIGHT key PAGE-LEFT key DELETE key NO key YES key

CONDITIONALS AND LOGICAL OPERATIONS CO CO object CTRL-G CTRL-Z IF condition list1 (list2) IFFALSE (IFF) list IFTRUE (IFT) list OUTPUT (OP) object PAUSE REPEAT number list RUN list STOP TEST condition

LOGICAL OPERATIONS AND condition1 condition2 ... NOT condition OR condition1 condition2 ...

INPUT OUTPUT COMMANDS CHARP FIREP number FPRINT object JOY number PRINT (PR) object ... READCHAR (RC) READLIST (RL) STARTPRINT STOPPRINT TYPE object1 ... WAIT time

WORKSPACE MANAGEMENT BURY package ERALL ERALL package(list) ER procedurename(s) ERN variablename(s) ERNS ERNS package(list) ERPS ERPS package(s) PACKAGE packagename procedurename(s) PKGALL package PO name(s) POALL POALL package(s) PONS PONS package(s) POPS POPS package(s) POTS POTS package(s) UNBURY UNBURY package

FILE COMMANDS CATALOG DISK ERASEFILE (ERF) filename.extension LOAD filename LOAD filename packagename LOADCHAR filename LOADSH filename LOADPICT filename SAVE filename SAVE filename package(s) SAVECHAR filename SAVESH filename SAVESH filename shapelist SAVEPICT filename SETDISK drivenumber

ADVANCED PROCEDURE DEFINITION COPYDEF newname name DEFINE name list PRIMITIVEP name TEXT name

MISCELLANEOUS DEFINEDP word PROCEDUREP name GO word LABEL word NODES RECYCLE REPARSE .CONTENTS .VEXAMINE N .VDEPOSIT N byte

LOGO SHORTCUT NOTES

CTRL+Y in LOGO will bring up the last line entered, this can save a lot of typing.

You can type "**CATALOG**" and there's a file you can load with **LOAD "EASYWAY** that has some circle and arc drawing routines.

To run the LOGO activities you can use these shortcuts (minimum 3 characters typed, but it must be unique to that shortcut). Note that shortcuts match the titles given in the main menu.

- **LOGO**
- **LOGO ACTIVITY HELP** (can type shortcut LOGO A) for instructions to the Logo activities.
- **LOGO REFERENCE** (can type shortcut LOGO R) for an online reference manual for Logo.
- **LOGO PREVIEW**
- **ALPHA LAB (L)**
- **MINERAL HUNT (L)**
- **O CANADA (L)**
- **PROVINCES (L)**
- **BEEVADERS (L)**
- **WHIZ MIND (L)**
- **ANGLE TANGLE (L)**
- **MATH ATTACK (L)**
- **HUNGRY DRAGON (L)**
- **MAZE CRAZE (L)**
- **NIMBLE (L)**

LOGO ACTIVITY HELP

READY TO GO? PRESS GO
NEED HELP? PRESS HELP
NEW CHOICE? PRESS EXIT
PREVIOUS MENU? PRESS RESTART

COPYRIGHT 1984
NABU NETWORK CORP.

LOGO ACTIVITY HELP PAGE 1 OF 2

USE THE JOYSTICK OR THE ARROW KEYS TO MOVE THE RED SQUARE OVER YOUR CHOICE, THEN PRESS THE FIRE BUTTON OR PRESS GO.

PREVIOUS PAGE OR NO PREVIOUS
RESTART, GO TO PREVIOUS MENU
EXIT TO NABU CATEGORY MENU
HELP
SWITCH TO TELEVISION
NEXT PAGE OR NO NEXT

LOGO ACTIVITY HELP PAGE 2 OF 2

TO PAGE THROUGH THE APPLICATION

NEXT PAGE: MOVE RED SQUARE TO RIGHT TRIANGLE THEN PRESS FIRE BUTTON OR GO

PREVIOUS PAGE: MOVE RED SQUARE TO LEFT TRIANGLE THEN PRESS FIRE BUTTON OR GO OR USE THE PAGE KEYS <| |> ON THE KEYBOARD
LEAVE HELP: RED SQUARE TO R, PRESS GO

LOGO ACTIVITY HELP

WHAT WOULD YOU LIKE TO SEE?

OVERVIEW
DARTS AND BALLOONS
MATH ATTACK
WHIZ MIND
MAZE CRAZE
ALPHA LAB
ANGLE TANGLE
O CANADA
MINERAL HUNT
PROVINCES

OVERVIEW PAGE 1 OF 3

THERE ARE A NUMBER OF PROGRAMS (ACTIVITIES) THAT ACCOMPANY THE NABU LOGO LANGUAGE. EACH PROGRAM WAS WRITTEN IN LOGO. THE PURPOSE OF THESE PROGRAMS IS:

- TO DEVELOP SKILLS
  COORDINATION
  LOGIC
  DEXTERITY
- TO CHALLENGE YOU
- TO SHOW YOU HOW TO PROGRAM IN LOGO.

OVERVIEW PAGE 2 OF 3

LET'S LOOK A BIT MORE AT PROGRAMMING IN LOGO. WE WANT TO ENCOURAGE YOU TO LOOK AT THE LOGO PROGRAM IN THE LOGO EDITOR. TO STOP A LOGO ACTIVITY, PRESS CTRL-G.

TO GO INTO THE LOGO EDITOR TYPE THE COMMAND "ED" FOLLOWED BY A SPACE, A DOUBLE QUOTATION MARK AND THE NAME OF THE PROCEDURE TO BE CHANGED.
ED "MAZE

OVERVIEW PAGE 3 OF 3

FOR EXAMPLE, TRY CHANGING THE SPEED OF THE CARS IN MAZE CRAZE, OR CHANGE THE CAR TO A TRUCK OR A FISH, WHATEVER YOU WISH!

YOU CAN RE-PROGRAM ANY PROGRAM TO DO WHAT YOU WANT IT TO. FEEL FREE TO EXPERIMENT.

DARTS AND BALLOOONS PAGE 1 OF 2

THE AIM OF THIS GAME IS TO DEVELOP YOUR ABILITY TO JUDGE SPEED AND DISTANCE. THE OBJECT OF THIS GAME IS TO STOP THE DART AS CLOSE TO THE BALLOON AS POSSIBLE.
PRESS THE FIRE BUTTON TO START THE DART MOVING. THE DART TRAVELS AT VARIOUS SPEEDS.

THE DART WILL DISAPPEAR AFTER ONE SECOND. PRESS THE FIRE BUTTON AGAIN TO STOP THE DART.

DARTS AND BALLOONS PAGE 2 OF 2

THE CLOSER THE DART IS TO THE BALLOON, THE MORE POINTS YOU GAIN. NO POINTS ARE AWARDED IF THE DART BREAKS THE BALLOON. IF THERE IS ONLY ONE PLAYER USE GAMES CONTROLLER 1.
IF THERE ARE 2 PLAYERS USE BOTH GAMES CONTROLLERS 1 AND 2. PLAYER WITH GAMES CONTROLLER 1 WILL BEGIN.

MATH ATTACK PAGE 1 OF 2

THE AIM OF THIS GAME IS TO PRACTISE YOUR ARITHMETIC SKILLS WHILE HAVING FUN.
CHOOSE YOUR LEVEL. THE HIGHER THE LEVEL THE FASTER YOU MUST ANSWER THE PROBLEM.
USE THE JOYSTICK TO MOVE THE BASE LEFT AND RIGHT.

MATH ATTACK PAGE 2 OF 2

PRESS THE FIRE BUTTON TO SHOOT THE ARITHMETIC SIGN NEEDED TO GIVE THE CORRECT ANSWER TO THE PROBLEM AT THE TOP OF THE SCREEN.
IF YOU SHOOT THE CORRECT SIGN BEFORE THE SPACESHIP LANDS, YOU WILL GET A CHANCE TO EARN MORE POINTS BY SHOOTING SMALLER SPACESHIPS.
SCORING:

LARGE SPACESHIP	20 POINTS
SMALL SPACESHIP	10 POINTS

WHIZ MIND PAGE 1 OF 2

THE AIM OF THIS GAME IS TO DEVELOP YOUR DEDUCTIVE REASONING. THE OBJECT OF THIS GAME IS TO TRY TO GUESS THE NUMBER THE COMPUTER HAS PICKED.
COLORED CIRCLES APPEAR ON THE RIGHT TO TELL YOU HOW NEAR YOUR GUESS IS.
THE CIRCLES MAY BE IN A DIFFERENT ORDER FROM THE NUMBERS.

WHIZ MIND PAGE 2 OF 2

BLUE CIRCLE	- WRONG DIGIT
YELLOW CIRCLE	- RIGHT DIGIT WRONG POSITION
RED CIRCLE	- RIGHT DIGIT RIGHT POSITION

YOU HAVE 9 GUESSES TO FIND THE NUMBER.

MAZE CRAZE PAGE 1 OF 2

THE AIM OF THIS GAME IS TO SHARPEN VISUAL PERCEPTION. THE OBJECT OF THE GAME IS TO MOVE THE CAR THROUGH THE MAZE TO GET HOME.
THERE ARE 3 MAZES OF VARYING DIFFICULTY TO CHOOSE FROM. PRESS THE FIRE BUTTON TO START THE CAR. USE THE JOYSTICK TO CONTROL THE CAR.
AFTER REACHING HOME, THE SPEED OF THE CAR IS INCREASED IN THE NEXT ROUND.

MAZE CRAZE PAGE 2 OF 2

IF YOU HIT THE WALL OF THE MAZE, THE
GAME STOPS. YOU MAY CONTINUE WITH THE SAME OPTION, OR CHOOSE A DIFFERENT MAZE.

THE CAR WILL STOP PERIODICALLY WHILE YOU ARE PLAYING THE GAME. THIS OCCURS WHEN LOGO PERFORMS A RECYCLE OR GARBAGE COLLECTION (CLEANING UP THE MEMORY). WHEN THIS HAPPENS, HOLD THE DIRECTION YOU WANT THE CAR TO MOVE IN UNTIL THE CAR STARTS MOVING AGAIN.

ALPHA LAB PAGE 1 OF 6
MATCHING CAPITAL
MATCH A CAPITAL LETTER TO A SMALL LETTER. LOOK AT THE CAPITAL LETTER NEXT TO ALPHA KID.
USE THE JOYSTICK TO MOVE THE CURSOR OVER THE SMALL LETTER THAT MATCHES THE CAPITAL LETTER. PRESS THE FIRE BUTTON. AFTER FIVE LETTERS, YOU RETURN TO ALPHA LAB OPTIONS.

ALPHA LAB PAGE 2 OF 6
MATCHING SMALL
MATCH A SMALL LETTER TO A SMALL LETTER IN THE SAME WAY AS IN "MATCHING CAPITAL".

ALPHA LAB PAGE 3 OF 6
ABC CAPITAL
PART OF THE ALPHABET WILL APPEAR IN CAPITAL LETTERS. THE CURSOR FLASHES WHERE A LETTER SHOULD BE. TYPE THE MISSING LETTER. IF THE LETTER IS CORRECT, THE CURSOR GOES TO THE NEXT BLANK. AFTER FIVE ROUNDS, YOU RETURN TO ALPHA LAB OPTIONS.

ALPHA LAB PAGE 4 OF 6
ABC SMALL
PART OF THE ALPHABET WILL APPEAR IN SMALL LETTERS THIS TIME. TYPE THE MISSING LETTERS IN THE SAME WAY AS IN "ABC CAPITAL".

ALPHA LAB PAGE 5 OF 6
SPELLING A
CHOOSE VOWELS TO COMPLETE WORDS. THREE PICTURES APPEAR WITH THEIR NAMES. THE VOWELS ARE MISSING. TYPE THE MISSING VOWEL. AFTER 12 PICTURES, YOU RETURN TO ALPHA LAB OPTIONS.

ALPHA LAB PAGE 6 OF 6
SPELLING B
SPELLING B USES THE SAME PICTURES AS IN SPELLING A. HOWEVER, NOW YOU MUST SPELL THE WHOLE WORD.

ANGLE TANGLE PAGE 1 OF 5
RECOGNIZE SHAPES AND LEARN ABOUT ANGLES AND DEGREES WITH ANGLE TANGLE. AN ANGLE IS FORMED BY TWO LINES MEETING AT ONE POINT. ANGLES ARE MEASURED IN DEGREES.

ANGLE TANGLE PAGE 2 OF 5
CIRCLE

THE OBJECTIVE IS TO LEARN THAT A CIRCLE MAY BE MEASURED IN DEGREES AND THAT THERE ARE 360 DEGREES TO A CIRCLE. THERE ARE THE SAME NUMBER OF DEGREES IN A LARGE CIRCLE AS WELL AS A SMALL CIRCLE. A FULL TURN IS ANOTHER KIND OF CIRCLE.

ANGLE TANGLE PAGE 3 OF 5
SQUARE - RECTANGLE
THE OBJECTIVE IS TO LEARN THE BASIC CHARACTERISTICS OF A SQUARE AND A RECTANGLE. THESE SHAPES CONTAIN 90 DEGREE ANGLES.

ANGLE TANGLE PAGE 4 OF 5
PROTRACTOR
THE OBJECTIVE IS TO MEASURE ANGLES USING A PROTRACTOR.
AN ANGLE WILL APPEAR BELOW THE PROTRACTOR. THE SAME ANGLE WILL BE DRAWN ON THE PROTRACTOR. READ THE PROTRACTOR, THEN TYPE IN THE SIZE OF THE ANGLE BEING MEASURED. THE ROUND IS OVER AFTER FIVE ANGLES HAVE BEEN ANSWERED CORRECTLY.

ANGLE TANGLE PAGE 5 OF 5
HOW TO ANSWER
WHEN A QUESTION APPEARS, TYPE YOUR ANSWER, THEN PRESS "GO".
IF YOU WOULD LIKE THE ANSWER PRESS "A".
THE NEXT QUESTION APPEARS AFTER YOU HAVE ANSWERED CORRECTLY OR HAVE PRESSED "A".

O CANADA PAGE 1 OF 3
THE OBJECTIVE IS TO CORRECTLY IDENTIFY THE PROVINCES AND TERRITORIES OF CANADA.
A MAP OF CANADA WILL APPEAR ON THE SCREEN. A FLASHING ARROW WILL POINT TO THE PROVINCE OR TERRITORY YOU ARE TO IDENTIFY.
THE LETTERS "A" TO "L" APPEAR BESIDE THE NAMES OF THE PROVINCES AND TERRITORIES BELOW THE MAP.

O CANADA PAGE 2 OF 3
TYPE THE LETTER BESIDE THE NAME OF THE PROVINCE OR TERRITORY POINTED TO BY THE FLASHING ARROW.
IF YOUR CHOICE IS CORRECT, THE PROVINCE OR TERRITORY IS FILLED WITH COLOR.
THE MAP IS COMPLETELY COLORED WHEN YOU HAVE IDENTIFIED THE TEN PROVINCES AND TWO TERRITORIES CORRECTLY.

O CANADA PAGE 3 OF 3
IF YOU PRESS A KEY OTHER THAN THE LETTERS "A" TO "L", THE ARROW POINTS TO THE NEXT PROVINCE OR TERRITORY.

MINERAL HUNT PAGE 1 OF 3
THE OBJECTIVE IS TO LEARN WHERE THE MAJOR MINERAL DEPOSITS ARE LOCATED IN EACH OF THE 10 PROVINCES OF CANADA.

TO CHOOSE THE PROVINCE YOU WISH TO EXPLORE, TYPE THE LETTER BESIDE THE NAME OF THE PROVINCE. TO AID YOU IN YOUR MINERAL HUNT, WE WILL DEPOSIT $2000 IN YOUR SAVINGS ACCOUNT.

MINERAL HUNT PAGE 2 OF 3

YOU CAN HUNT FOR MINERALS BY MOVING THE MINER WITH THE JOYSTICK. WHEN YOU HAVE DECIDED WHERE TO DIG, PRESS THE FIRE BUTTON.

YOUR SAVINGS ACCOUNT WILL INCREASE WITH EACH SUCCESSFUL DIG AND DECREASE WITH EACH UNSUCCESSFUL DIG. YOU WILL ALSO LOSE MONEY WITH EACH STEP THAT THE MINER TAKES.

MINERAL HUNT PAGE 3 OF 3

THE MINERAL HUNT ENDS WHEN YOU HAVE FOUND ALL THE MINERAL DEPOSITS IN A PROVINCE OR HAVE LOST ALL YOUR MONEY.

YOU WILL THEN RETURN TO MINERAL HUNT OPTIONS TO CHOOSE ANOTHER PROVINCE TO EXPLORE.

GOOD LUCK!

PROVINCES PAGE 1 OF 4

THE OBJECTIVE IS TO IDENTIFY THE 10 PROVINCES OF CANADA AND THEIR CAPITALS, AS WELL AS TO DISCOVER THE LOCATION OF THEIR CAPITALS.

A MAP OF ONE OF THE PROVINCES WILL APPEAR ON THE SCREEN. CORRECTLY IDENTIFY THE PROVINCE BY USING THE JOYSTICK TO HIGHLIGHT ITS NAME FROM THE LIST OF PROVINCES ON THE RIGHT; THEN, PRESS THE FIRE BUTTON.

PROVINCES PAGE 2 OF 4

WHEN YOU HAVE ANSWERED CORRECTLY, A LIST OF PROVINCIAL CAPITALS WILL APPEAR. IDENTIFY THE CORRECT CAPITAL BY USING THE JOYSTICK TO HIGHLIGHT YOUR CHOICE FROM THE LIST; THEN, PRESS THE FIRE BUTTON.

AFTER CHOOSING THE CORRECT CAPITAL, YOU WILL BE GIVEN A HELICOPTER TO FLY TO THE LOCATION OF THE CAPITAL ON THE MAP.

PROVINCES PAGE 3 OF 4

USE THE JOYSTICK TO FLY THE HELICOPTER. WHEN YOU DECIDE WHERE THE CAPITAL IS LOCATED, PRESS THE FIRE BUTTON.

YOUR FUEL DECREASES WHILE YOU ARE SEARCHING FOR THE CAPITAL, AS WELL AS, WHEN YOU MAKE AN INCORRECT LANDING.

PROVINCES PAGE 4 OF 4

THE ROUND ENDS WHEN YOU HAVE DISCOVERED THE LOCATION OF THE CAPITAL OR YOUR FUEL RUNS OUT.

A NEW ROUND WILL NOW BEGIN.