NeXT Logo

NeXT Logo interface Created by:
Gary F. Desrochers and Sean Kerstiens
for: Professor Aaron Gordon

Background:

Logo is a computer programming language that was initially intended to encourage children to learn about computers by creating shapes that are drawn by moving a turtle around a computer screen. Although the logo environment focuses on graphics and the graphical interface, it is also a powerful programming language.

NXLogo was designed at Colorado School of Mines during a six week summer field session for undergraduate students. We, Gary Desrochers and Sean Kerstiens, undertook the task of creating a working logo interface for the Next by the end of the course. The original logo interface had to be modified and a NeXT interface had to be added. The result of these efforts is NXLogo. Currently, the version is 1.0, as seen in the info panel of the application.

Differences and Modifications:

In general, most of the regular logo commands work for NXLogo. A few exceptions and changes are listed below. At the end of this file, a complete list of commands is given.

There are some parsing differences between logo and NXLogo. To begin with, a left bracket ([) must be closed on the same line as it is opened. This can be seen using a print statement. Here is a correct statement:

print [Hello. How are you?]

Also, there are no multi-line commands. This feature could not be added in the time period that was provided. It a useful feature that could be added in the future. There are also a couple new ways to enter a <to> procedure. To define a procedure from within the NXLogo application type:

to <procedurename> <:variable(s)> | <procedure must be on one line>

Another way to make a procedure is to create a file using a text editor. (Once you have defined a procedure from within Logo, this is the only way to make changes to that procedure. Unless you want to erase it and then redefine it.):

to <procedurename> <:variable(s)>
<procedure in as many lines as you want
end

In version 1.0 of NXLogo the procedure has to start on the first line of the file for NXLogo to recognize it properly. Once you have made the file in the editor, the file must be saved as <procedurename>.lg in your working logo directory.

The working logo directory is another important point. There is a system library (included in the package as NXLogoLibrary) that can hold any of the global (everyone can use) logo procedures. NXLogo will search your home directory first before it goes and does the main library search. If it doesn't find the procedure in there, then it will tell you it doesn't know how to do that procedure name. This means that you can redefine a system defined procedure by making your own. You can change your personal working directory to anything you like with a NeXT unix command called dread. To do this simply perform a dread NXLogo NXLogoLib <pathname terminated with a />. Here is and example:

dwrite NXLogo NXLogoLib <Net/Users/tsmith/logo.library/>

It is very important to include the slash (/) after the last directory name, otherwise NXLogo will not recognize the directory. The system library can also be changed with this command. There is a help directory (included in the package as NXLogoLibrary/help) and this can be changed with the dread also. These two can be changed with:

dwrite NXLogo NXLogoSysLib <Net/user/lib/logo.library/>
dwrite NXLogo NXLogoHelp <Net/user/help/logo.help/>

In the future this will also be made into a preferences menu for the system administrator so that he will not have to relink the whole program to change the directory. This would not be too hard but was not done due to time limitations.

Another way to personalize your environment is to drag the NXLogo program's icon to the right side of the screen under the Next Cube and then every .logo file will show up as a turtle. In the future, when a .logo file is double clicked NXLogo will load that one file and display it in the NXLogoprogram. Currently it only loads up logo. (Sorry folks, maybe in another version!)

The following are new features built into the graphical interface of NXLogo.

(Version 1.0)

Menu options:

Main menu:

Info...:

It gives you the authors of the graphical interface. (who also had to do the modifications to the non-graphical interface.

File:

New:

This does the same as a clearscreen command within NXLogo. It will clear the line buffer, center the Turtle and start with a clean state. New does not change the location or direction of the turtle.

Open:

This will give you a promt for what file you want to load and starts you from where that file had left off. This command does a new before it loads up the file.

Add:

This will prompt you for a what file you want to add and starts you from where that file had left off. This command does not do a new before it loads up the file. It just puts this file on top of what you already have.

Save:

This menu option saves the file to the last file you accessed. If you have not accessed a file then it will bring up a open dialog box that will allow you to enter the name you wanted to use.

Save As...:

The Save As is the same as the Save except for it always asks you for the file to access.

Save To...:

The Save To is the same as the Save except for it always asks you for the file to access. Later it will allow for multiple formats to save the file in.

Format:

This contains only Page layout as a menu off of it and it does everything that a normal Page layout does.

Edit:

This contains all the normal editing features that can be done within the text object.

Print:

This menu command allows for you to print out the graphics screen (minus the turtle). It brings up the print dialog box and asks you the normal stuff.

Hide:

This hides the NXLogo program.

Quit:

This menu option quits NXLogo. The same can be done typing out the logo command bye or using a <command key><q>.

The Following are the commands that are available within NXLogo. If there are any changes to the documented call (or online help) then these will be listed after each command. Some commands have another small alias that can be used instead of the full commands and these are listed after each command. The format is:

<*Full Command Name***> , <***Abbreviated Command Name***> : <***Differences***>**

***** **WARNING** *****

Most of these commands have been used in the NeXT environment. Some of the commands, and all the different methods of use have not been tried. This warning only applies to the more obscure procedures. If you have any problems then please contact the NXLogo interface authors.

sum,add
difference,diff
product
quotient
remainder,mod
both,and
either,or
greaterp
lessp
maximum,max
minimum,min
is
word
sentence,se
equalp

emptyp
not
numberp
sentencep
wordp
count
zerop
first
butfirst,bf
last
butlast,bl
thing
namep
random,rnd
sqrt
pow
sin
cos
arctan,atan
time
print,pr
type
output,op
wait,
local
make
if
to : see the above documentation about how to use this command.
end
stop
break
go
show,po
erase,er : remember erase only erases yours not the system.
help
describe
trace
untrace
goodbye,bye: This also quits from the whole program.
turtle,tur
forward,fd
back,bk
left,lt
right,rt
penup,pu
pendown,pd

clearscreen,cs: This command will clear the line buffer.
Maybe later on there can be added also a
remove $ of line(s) command
showturtle,st
hideturtle,ht
penerase,pe
pencolor,penc
setcolor,setc
setxy
setheading,seth
wipeclean,clean
penmode
penreverse,px,
shownp
towardsxy
xcor
ycor
heading
getpen
scrunch
setscrunch,setscrun
toplevel
fprint,fp
ftype,fty
pots: Maybe later a new command can be added to print out the system library.
fput
lput
list
openread,openr
openwrite,openw
close
fileread,fird
fileword,fiwd
fileprint,fip
filefprint,fifp
filetype,fity
fileftype,fifty
unix: Does not allow for interactive unix commands(like vi). Only allows
for non-interactive commands.(ls, cat, fgrep, ect.)
run
repeat
repcount
cbreak
int
round
ascii
char

oflush
gprop
plist
pprop
pps
remprop
test
iftrue,ift
iffalse,iff
memberp
item,nth

We hope you enjoy this program. Any comments (Yes we do take comments, good, bad, or otherwise) then send them to:

gdesroch@slate.mines.colorado.edu
skerstie@slate.mines.colorado.edu
agordon@mines.colorado.edu
or
Contact the Department of Mathematical and Computer Sciences at Colorado School of Mines, Golden, Colorado, 80401, 303-273-3860.

NeXT Logo

NeXT Logo interface Created by:
Gary F. Desrochers and Sean Kerstiens
for: Professor Aaron Gordon
