

PC LARN version 12.3 for IBM PC Compatibles

Table of contents

1. Introduction
2. System requirements
3. Files Supplied
4. Installation
5. Configuration
6. Command line options
7. TERMCAP
8. WIZARD mode
9. History and other Information

1. Introduction

LARN is a dungeon type adventure game similar in concept to HACK, ROGUE or MORIA, but with a different feel and winning criteria. LARN was released for the UNIX environment in 1986 by Noah Morgan. It was subsequently ported to the MS-DOS environment by Don Kneller. Kevin Routley has been working on enhancements to LARN on and off for the past two years.

2. System requirements

PC LARN requires:

- MS-DOS or PC-DOS 2.x or above.
- at least 256K of RAM.
- disk storage capacity of at least 360K, although twice that much is preferred if you want to "checkpoint" your game to prevent accidental lose due to a system failure.
- the ANSI.SYS device driver must be installed (although the NANSI.SYS driver is *strongly* preferred and has been included in this distribution). You should install either ANSI.SYS or NANSI.SYS, but not both. To install NANSI.SYS, put the line "device=NANSI.SYS" in your CONFIG.SYS file. See the TERMCAP section for further details.

3. Files supplied

The following files should be included in this LARN123.ZIP file:

LARN123.DOC	This documentation.
LARN123.FIX	Changes made between LARN V12.2 and V12.3.
LARN123.EXE	The executable file.
LARN.FTN	Fortune cookie messages.
LARN.HLP	A help file, which can be read with the "?" command in LARN.
LARN.MAZ	Some maze levels are pre-calculated and stored here.
LARN.OPT	An example LARN.OPT file (see the CONFIGURATION section).
TERMCAP	A file describing the escape sequences to send to the terminal. The supplied TERMCAP file is for monochrome systems. See the TERMCAP section for details on changing TERMCAP.
NANSI.DOC	The documentation for NANSI.SYS.
NANSI.SYS	A new ANSI.SYS that is *much* faster than ANSI.SYS and supports the line insert and line delete escape sequence that LARN uses. This excellent product is written by Daniel Kegel. See the TERMCAP section and NANSI.DOC for more details.

4. Installation

NANSI.SYS or ANSI.SYS must be installed as a device driver on your system in order for LARN to work (see System Requirements above).

For a hard disk system I suggest creating a \GAMES\LARN directory and placing LARN and the provided support files in that directory.

Following that, you would place \GAMES\LARN in your PATH.

LARN will create several other files (LARN.SCR, LARN.PID, LARN.LOG) in that directory. Also, save files (LARN.SAV) and checkpoint files (LARN.CKP) will go in there by default. You will want to set the 'larndir:' option in your LARN.OPT file (see the Configuration section below) to point to this directory.

For two-floppy systems, I would suggest the following:

- o place LARN123.EXE, LARN.OPT and TERMCAP. on disk one.
- o place LARN.FTN, LARN.MAZ, and LARN.HLP on disk two. This will be the 'larndir:' disk (see the Configuration section below). LARN.SCR, LARN.PID, LARN.LOG, save files, and checkpoint files will go on this disk.

5. Configuration

PC LARN is configured by options placed in the file LARN.OPT. When PC LARN starts up, it looks in the current directory for this file, then in directories along your PATH. Here are the options that can be put in LARN.OPT.

Options that are followed with a colon, ":", take at least one argument. Options without a colon are boolean, meaning you select that option merely by mentioning it.

Comments can be entered in the options file by using the '#' character.

These options are only in PC LARN:

cursor: start-line end-line
Normally, LARN uses the standard DOS cursor (the flashing underscore) to show where your player is located. This can be difficult to see. This option results in a *BIOS* call to change the cursor size. For a monochrome display, the normal value of start-line and end-line is 11 and 12. Values of 4 and 9 give a larger cursor which is not too obtrusive.
If "cursor" is not specified, no BIOS call is made.

DECRainbow
Put this in LARN.OPT if your computer is a DEC Rainbow and you want to use graphics characters for drawing the maze. You can use any characters from the ASCII character set and from the "line-drawing" character set. To use line-drawing characters, take the decimal value of the character and add 128 to it. This makes the values of line-drawing characters lie between decimal 128 and 255 and tells PC LARN you want a line-drawing character.

Every attempt has been made with PC LARN V12.3 to maintain DECRainbow compatibility as provided by Don Kneller.

Unfortunately, since I do not have access to a Rainbow, I cannot make any guarentees.

graphics: wallc floorc
In UNIX LARN, the wall character is a # and the floor character is a blank (which makes it almost impossible to tell where you've been). In PC LARN, the default wall character is also a # but the default floor character is a period.

This option allows you to change these characters to something you like. Suggested characters are 176 (a graphic block) and 249 (a central dot).

Almost all ASCII (non-graphics) characters already represent objects in LARN, so you should only choose numbers that are in the graphics character set (i.e. those with decimal values greater than 128).

keypad
This enables the keypad for use with LARN. A *BIOS* call is used to read the keypad. The correspondance between keypad and game command is:

Keypad	Command
7 8 9	u+l(y) up(k) u+r(u)
\ /	\ /
4 -5- 6	left(h) nothing right(l)
/ \	/ \
1 2 3	d+l(b) down(j) d+r(n)
Ins	inventory(i)

The letter in () is the game command letter. "u+l" means up and left. The keypad "7" is translated to the command for moving one space up and left. With shift the commands are the same as the uppercase letters.

larndir: directory
Sets "directory" to be the place LARN looks for files. For hard disks set this to be \GAMES\LARN. For 2-floppy systems, use B:

If "larndir" is not specified, the current directory is assumed.

ramlevels: number
This number is the maximum number of levels PC LARN will keep in

LARN memory. The number can be between 1 and 14. Normally, PC will use as much memory as available, but you may want to use less so there will be enough memory to start up a second copy of COMMAND.COM with the LARN command `!'. If there isn't enough memory for COMMAND.COM, LARN will clear the screen and prompt with "A>" until return is struck, then go back to the game. If "ramlevels" is not specified, a value of 14 is used. If "number" is less than 14, LARN will use a swapfile to store the oldest levels. Note that a level takes about 8K of memory.

rawio This option causes the input and output of your computer to be switched to "raw" mode. The advantage of this is faster output and better handling of special characters ^S and ^P.

```
===== WARNING =====  
Some computers (eg. DEC Rainbows) hang if rawio is attempted.  
=====
```

swapfile: filename
This is the name of the swap file that will be used if ramlevels is less than 14. If all the levels fit in memory, no swap file will be used. You may want to put the swapfile on a RAMdisk. If "swapfile" is not specified, LARN.SWP in directory larndir will be used.

The following are options usable with UNIX LARN as well:

auto-pickup
player Starts you out in 'automatic pickup' mode. Whenever the moves onto an object, it will be picked up. This mode can be toggled with the '@' command while in the game.

bold-objects
objects' By specifying this option, all objects in the game will be highlighted. The 'highlight-objects' and 'inverse-objects' specify which highlighting method will be used. The default is non-bold objects, except if the 'original-objects' option is specified, when it is *forced* on (to distinguish monsters and objects that are represented by the same character).

enable-checkpointing
With this option, the game is saved every 400 moves. If your

system crashes it is possible to recover from the last
checkpoint file (LARN.CPK in the LARN directory) with the command "larn
++".

highlight-objects

If the 'bold-objects' option is specified, this option causes
objects to be highlighted with bolding.

inverse-objects

If the 'bold-objects' option is specified, this option causes
objects to be highlighted with inverse video. The default is
'highlight-objects'.

monster: name

Sets the name of a monster. If the name has a space in it,
enclose it in double quotes (eg "tooth fairy"). The leading
character determines which monster the name goes with (e.g.
"monster: bee" and "monster: Buzzbomb" change the names of
bugbears and Bats, respectively).

name: yourname

Sets the name of the player. If you want a space in your
name,
enclose it in double quotes (eg "Mad Max").

no-introduction

Skip displaying the opening messages.

no-beep

Disable sound from the speaker.

original-objects

Like 'prompting' mode, this mode is provided for
compatibility
with LARN V12.0. By specifying this option, the characters
used to show objects in the game are the same as in 12.0.
The
with
default is to use characters that are closer to those used
HACK, ROGUE and MORIA.

prompt-on-objects

By specifying this option, you enable 'prompting' mode. In
this
mode, the player is prompted for the action to perform when
encountering an object (e.g. 'Eat, pick up, or ignore?').
Prompting mode is provided for compatibility with LARN V12.0,
and
is off by default.

savefile: filename

The filename to use for saving the game. The default is
LARN.SAV
in the LARN directory. For a 2-floppy system you might try
B:LARN.SAV if the LARN directory is on the A disk drive.

6. Command line options

There are several command line options that can modify the behavior of LARN. These are:

-o optionfile Use this as the option file rather than LARN.OPT. As with LARN.OPT, the current directory then directories along your path are searched for this file which supplies configuration information.

-s Show scores.

-l Show log file. You have to know the Wizard's password to do this.

-i Show all scores including the inventories of dead players.

-c Create a new score file. You have to know the Wizard's password to do this. Erasing LARN.SCR has the same effect.

-n No welcoming message. Prevents printing of the short introduction" effect. Putting the "no-configuration option in LARN.OPT has the same effect.

-# Where # is a number from 0 to 9. This sets the difficulty of LARN to this level. Normally, LARN starts out with difficulty 0 and increases in difficulty by 1 when you win at the current level. Thus the game automatically gets more difficult.

-h, -? A help screen that shows the command line arguments.

-p Prompt for actions on objects. Specifying the 'prompt-on-objects' option in LARN.OPT has the same affect.

++ Restore a game from a checkpoint file. If you have checkpointing enabled (with the "enable-checkpointing" a system file crash you can recover the game from the checkpoint with this command line option

For example: "larn -n -o bobslarn.opt" starts up LARN with no introductory message and uses bobslarn.opt file for configuration options. "larn ++" tries to restore LARN from a checkpoint file.

7. TERMCAP

LARN uses the UNIX "termcap" to select escape sequences to be sent to the terminal driver (either ANSI.SYS or NANSI.SYS) to change video modes.

When LARN starts up, it checks in the environment for the variable called

"TERM". The DOS command:

```
set TERM ibmpc-ega
```

will give the environment variable "TERM" the value "ibmpc-ega". By default, LARN assumes the value of TERM is "ibmpc-mono".

Then LARN looks in the file called "TERMCAP" for a termininiefinition

with the same name as the value of TERM. LARN first looks for TERMCAP in

the current directory, then in directory "\ETC" (the normal UNIX place),

then in directories along your PATH.

The TERMCAP file uses 2 letter codes to describe each escape sequence

and \E to mean ESC (the escape character is decimal 27). Each entry is

enclosed in colons. For example, to start underscore mode on a monochrome

display, we want to send the sequence "ESC[4m". This looks like ":us=\E4m:" as a termcap entry.

The following sequences are used by LARN:

```
ti terminal initialization. You could have something like:
:ti=\E44;37m:
```

which would mean to select blue background and white foreground.

```
te terminal end. To reset your terminal to white on black:
:te=\E0m:
```

so stand out. Select red forground, keeping the blue background:

```
:so=\E31m:
```

```
se standout end. Back to white on blue:
:se=\E44;37m:
```

al insert line. This is *not* available in ANSI.SYS, so remove this entry if you're using ANSI.

dl delete line. This is *not* available in ANSI.SYS, so remove
this entry if you're using ANSI.

If you want to make your own TERMCAP entry, copy the "ibmpc-mono"
entry, replace the name (eg use "ibmpc-color" or "ibmpc-ega" or
whatever),
then replace the sequences with whatever you please. The possible
escape sequences are generally listed in the DOS manual in the description
of ANSI.SYS, as well as in the NANSI.DOC file.

Remember to change the environment variable "TERM" to reflect which
TERMCAP entry you want! You should probably put a "set TERM=..."
line in your AUTOEXEC.BAT file.

8. WIZARD mode

There is a WIZARD mode for testing features of the game. To get
into WIZARD mode, type in an underscore "_" and answer the prompt for the
password with "pvnert(x)" (do not enter the quotes). Wizards are
non-scoring characters that get enlightenment, everlasting expanded
awareness and one of every object in the game.

9. History and Other Information

Noah Morgan originally created LARN 12.0 and released the UNIX
version to the USENET in 1986. Don Kneller ported the UNIX
version to MSDOS (both IBM PCs and DEC Rainbows).

Kevin Routley has been working on LARN enhancements on and off for
the past two years. Version 12.1 had a limited distribution.
Version 12.2 was distributed to the Usenet community. Version 12.3
may be last version from Kevin that will maintain savefile
compatibility with version 12.0 savefiles. Some future version will be released
as Shareware.

Other editions of Larn have been distributed by others, namely
LARN13 and Ultra-Larn.

I hope you enjoy this version of LARN. Please send any questions,
suggestions, requests, or comments regarding LARN and LARN V12.3
in particular to:

Kevin Routley
9 Aster Court
Nashua, NH 03062
ARPA: routley@tle.dec.com
BITNET: routley@tle.dec.com

March 23, 1991