NAME

frotz − interpreter for Infocom and other Z-Machine games

SYNOPSIS

frotz [*options*] *file* [*blorb_file*]

At least one file must be specified on the command line. This can be either a plain Z-code file or a Blorb file. A Z-code file is a compiled executable for the Z-Machine. A Blorb file contains audio, graphics, and other things in addition to the game wrapped up into a single file. It can also optionally contain the Z-Machine executable. If a plain Z-code file is supplied, then**Fr otz** will check for a Blorb file with the same base name but an extension of **.blb, .blorb,** or **.zblorb** and load it if found.

If the file supplied on the command line is a Blorb file, then **Frotz** will check to see if a Z-code file is contained within. If not found, then **Frotz** will complain and exit.

An alternatively-named Blorb file can be supplied as the optional second parameter to the command line invocation.

DESCRIPTION

Frotz is a Z-Machine interpreter. The Z-machine is a virtual machine designed by Infocom to run all of their text adventures. It went through multiple revisions during the lifetime of the company, and two further revisions (V7 and V8) were created by Graham Nelson after the company's demise. The specification is now quite well documented; this version of Frotz supports version 1.0.

This version of Frotz fully supports all these versions of the Z-Machine except for version 6. Version 6 is semi-supported by displaying the outlines of V6 graphics with the picture number in the bottom-right corner.

OPTIONS

−a Watch attribute setting. Setting and clearing of attributes on objects will be noted in debugging messages.

−A Watch attribute testing. Every time the Z-machine tests an attribute value, the test and the result will be reported.

−b <colorname>
Sets the default background color. <colorname> corresponds to one of the Z-machine colors, which are as follows:
black red green yellow blue magenta cyan white
If color support is disabled or not available on your terminal, this option does nothing.

−c N Sets the number of context lines used. By default, after a ''[MORE]'' prompt, and assuming there is enough output pending, Frotz will allow all the currently visible lines to scroll off the screen before prompting again. This switch specifies how many lines of text Frotz will hold over and display at the top of the next screen.

−d Disable color.

−e Enable sound. If you've disabled sound in a config file and want to hear sound effects, use this.

−E <mode>
Emphasis mode. <mode> corresponds to one of three possible ways to render emphasized text. Possible choices are as follows:
italic underline none
Infocom created an EMPHASIS_STYLE for the Z-machine, which was supposed to make the text be underlined or be italicized. Exactly what to do is left up to the interpreter. Some platforms/terminals could support either, both of these, or neither. Underlining was more common than italic. This option allows you to choose how EMPHASIS_STYLE is displayed.

−f <colorname>
Sets the default foreground color. <colorname> corresponds to one of the Z-machine colors, which are as follows:
black red green yellow blue magenta cyan white
If color support is disabled or is not available on your terminal, this option does nothing.

−F Force color mode. If you've disabled color in a config file and want to Frotz to display colors, use this.

−h N Manually sets the text height. Though most curses libraries are intelligent enough to determine the current width from the terminal, it may sometimes be necessary to use this option to override the default.

−i Ignore fatal errors. If a Z-Machine interpreter encounters a zcode error such as division-by-zero or addressing an illegal object, the proper response is to abort execution. This is done because the zcode program doesn't have a clear idea of what is going on. There are some games out there that cause fatal errors because the authors were careless and used an interpreter that didn't properly check for errors. This option is intended to get around such bugs, but be warned that Strange Things may happen if fatal errors are not caught.

−I N Set the interpreter number. Infocom designed the Z-machine such that a game could tell on what kind of machine the interpreter was running. See INTERPRETER NUMBER below.

−l N Sets the left margin, for those who might have specific formatting needs.

−L <filename>
When the game starts, load this saved game file.

−m Enable mouse support. Naturally, this is quite limited, but it's available for anyone who wants to experiment with it. When active, the mouse cannot be used to copy text from the terminal.

−o Watch object movement. This option enables debugging messages from the interpreter which describe the moving of objects in the object tree.

−O Watch object location. These debugging messages detail the locations of objects in the object tree.

−p Plain ASCII output only. This inhibits the output of accented letters and other characters from the Latin-1 character set, replacing them with reasonable alternatives. This may be necessary on devices lacking these characters. The OE/oe dipthongs are missing from the Latin-1 set. These are

handled as well.

−P Alter the piracy opcode. The piracy opcode was never used by Infocom. This switch is really only useful for those who like to toy around with Z-code.

−q Quiet. Turns off sound effects. Useful when running Frotz on a remote machine and you don't want to bother whoever's near the console with weird noises.

−r N Sets the right margin.

−R <path>
Restricted read/write. Reading and writing files will be restricted only to the provided path. Ordinarily Frotz will write or read its saves, transcripts, and move recordings in whatever path or directory the user provides when the **SAVE, SCRIPT,** or **RECORDING** commands are given. This can be undesirable if Frotz is run in a restricted environment, by a front end, or by a chatbot. This option will cause Frotz to write or read only to the provided path and nowhere else. Then the controlling process can then watch that directory for changes and need not worry about someone scribbling or snooping who-knows-where.

−s N Set the random number seed value. The given seed value is used as the initial seed value on every restart. This is helpful for testing games like **Curses** which make random decisions before the first input (such that the hot key Alt−S does not really help).

−S N Set the transcript width. By default your transcript files are formatted to a width of 80 columns per line, regardless of the current text width. This switch allows you to change this setting. In particular, use −S 0 to deactivate automatic line splitting in transcript files.

−t Sets the Z-machine's *Tandy bit*, which may affect the behavior of certain Infocom games. For example, Zork I pretends not to have sequels, and Witness has its language toned down.

−u N Sets the number of slots available for Frotz's multiple undo hotkey (see below). This defaults to twenty, which should be sufficient for most purposes. Setting too high a number here may be dangerous on machines with limited memory.

−w N Manually sets the text width.

−x Expand the abbreviations "g", "x", and "z" to "again", "examine", and "wait". This switch is for use with old Infocom games that lack these common abbreviations which were introduced in later games. Use it with caution: A few games might use "g", "x" or "z" for different purposes.

−v Show version information and exit. This will display the version of Frotz, some information about what's enabled and what's not, the commit date of the source code, and a git(1) hash of that commit.

−Z N Error checking mode.
0 = don't report errors.
1 = report first instance of an error.
2 = report all errors.
3 = exit after any error.
Default is 1 (report first instance of an error).

HOT KEYS

These hot keys are enabled only when the Z-machine is waiting for line input (for Z-machine experts: the @read opcode).

Alt-D Set debugging options.

Alt-H Help (print the list of hot keys).

Alt-N New game (restart).

Alt-P Playback on.

Alt-R Recording on/off.

Alt-S Set random number seed.

Alt-U Undo one turn.

Alt-X Exit game (after confirmation).

INTERPRETER NUMBER

The interpreter number is a setting in the Z-machine header which is used to tell the game on what sort of machine the interpreter is running. Frotz will automatically choose the most appropriate number for a given Infocom-produced game. Should you want to override the number, the **−I** option is available.

An interpreter should choose the interpreter number most suitable for the machine it will run on. In Versions up to 5, the main consideration is that the behaviour of 'Beyond Zork' depends on the interpreter number (in terms of its usage of the character graphics font). In Version 6, the decision is more serious, as existing Infocom story files depend on interpreter number in many ways: moreover, some story files expect to be run only on the interpreters for a particular machine. There are, for instance, specifically Amiga versions. The DECSystem-20 was Infocom's own in-house mainframe.

For Infocom's four V6 games, the interpreter number will be automatically chosen based on the title and release number. Of course, this can be overridden at the command line.

Infocom used the following interpreter numbers:

1 DECSystem 20

2 Apple IIe

3 Macintosh

4 Amiga

5 Atari ST

6 IBM PC

7 Commodore128

8 Commodore64

9 Apple IIc

10 Apple IIgs

11 Tandy Color

CONFIGURATION FILES

On startup, **frotz** will first check the system's frotz.conf then $HOME/.frotzrc for configuration information. The configuration file uses a simple syntax of
<variable> <whitespace> <value>

Color names may be any of the following:
black | red | green | blue | magenta | cyan | white

ascii on | off
Use plain ASCII only. Default is "off".

background <colorname>
Set background color. Default is terminal's default background color.

color yes | no
Use color text. Default is "yes" if supported.

errormode never | once | always | fatal
Set error reporting mode.
never Don't report any errors except for fatal ones.
once Report only the first instance of an error.
always Report every instance of an error.
fatal Abort on any error, even non-fatal ones.
Default is "once".

expand_abb on | off
Expand abbreviations. Default is off. Expand the abbreviations "g", "x", and "z" to "again", "examine", and "wait". This switch is for use with old Infocom games that lack these common abbreviations which were introduced in later games. Use it with caution. A few games might use the "g", "x", or "z" for different purposes.

foreground <colorname>
Set foreground color. Default is terminal's default foreground color.

ignore_fatal on | off

Ignore fatal errors. If a Z-Machine interpreter encounters a zcode error such as division-by-zero or addressing an illegal object, the proper response is to abort execution. This is done because the zcode program doesn't have a clear idea of what is going on. There are some games out there that cause fatal errors because the authors were careless and used an interpreter that didn't properly check for errors. This option is intended to get around such bugs, but be warned that Strange Things may happen if fatal errors are not caught.
Default is "off"

piracy on | off
Alter the piracy opcode. Default is off. The piracy opcode was never used by Infocom. This option is only useful for those who like to toy around with Z-code.

randseed <integer>
Set random number seed. Default comes from the Unix epoch.

sound on | off
Turn sound effects on or off. Default is "on".

tandy on | off
Set the machine's *Tandy bit*. This may af fect the behavior of certain Infocom games. For example, Zork I pretends not to have sequels, and Witness has its language toned down. Default is "off".

undo_slots <integer>
Set number of undo slots. Default is 500.

zcode_path /path/to/zcode/files:/another/path
Set path to search for zcode game files. This is just like the $PATH environmental variable except that you can't put environmental variables in the path or use other shortcuts. For example, "$HOME/games/zcode" is illegal because the shell can't interpret that $HOME variable.

The following options are really only useful for weird terminals, weird curses libraries or if you want to force a certain look (like play in 40-column mode).

context_lines <integer>
Set the number of context lines used. By default, after a ''[MORE]'' prompt, and assuming there is enough output pending, frotz will allow all the currently visible lines to scroll off the screen before prompting again. This switch specifies how many lines of text frotz will hold over and display at the top of the next screen. Default is "0".

left_margin <integer>
Set the left margin. This is for those who might have special formatting needs.

right_margin <integer>
Set the right margin. This is for those who might have special formatting needs.

text_height <integer>
Manually set text height. Most curses libraries are intelligent enough to determine the current width of the terminal. You may need to use this option to override the default.

text_width <integer>

Manually set text width. Again, this should not be necessary except in special circumstances.

script_width <integer>
Set the transcript width. Default is 80 columns per line, regardless of the current text width. This switch allows you to change this setting. You may set this to "0" to deactivate automatic line-splitting in transcript files.

The following options are mainly useful for debugging or cheating.

attrib_set on | off
Watch attribute setting. Setting and clearing of attributes on objects will be noted in debugging messages. Default is "off"

attrib_test on | off
Watch attribute testing. Every time the Z-machine tests an attribute value, the test and the result will be reported. Default is "off".

obj_loc on | off
Watch object location. These debugging messages detail the locations of objects in the object tree. Default is "off".

obj_move on | off
Watch object movement. This option enables debugging messages from the interpreter which describe the movement of objects in the object tree. Default is "off".

COLOR

Whether or not **Frotz** will display color depends upon the curses library and the terminal. In general, an xterm or other X11-based terminal emulator will support color. Sometimes the value of $TERM will need to be set to something like "xterm-color" or "rxvt-256color". For a Linux console, $TERM is almost always set to "linux". This will support color. For a NetBSD or OpenBSD console on an x86 or amd64, the default value of $TERM is "vt100". To get color supported there, you need to set $TERM to "pc3". A FreeBSD console's $TERM is "xterm" and will support color. Color on text consoles on machines other than x86 or amd64 is untested.

On some operating systems, Xterm will not change the cursor color to match that of the text. To fix this, add the following line to your **.Xresources** file and type **xrdb -merge $HOME/.Xresources**

xterm*cursorColor: *XtDefaultForeground

This can also be added to a systemwide file such as **/etc/X11/Xresources/x11-common** or **/etc/X11/app-defaults/XTerm.** The names and locations of the system-wide files can vary from OS to OS.

UNICODE

Frotz supports Unicode glyphs by way of UTF-8 if the terminal used supports UTF-8. If you prefer using xterm, start it as uxterm. This is a wrapper script that sets up xterm with UTF-8 locale. You can also manually tell an xterm to switch into UTF-8 mode by holding CTRL and the right mouse button to bring up the VT FONTS menu. Depending on how xterm was installed, you may see an option for "UTF-8 Fonts" which will allow Unicode to be properly displayed.

Getting normal xterm to behave like this all the time can vary from system to system. Other terminal emulators have their own ways of being set to use UTF-8 character encoding.

NON ASCII CHARACTERS

Non-ASCII glyphs can be displayed without the use of UTF-8 by way of the ISO-8859-1 or ISO-8859-15 (Latin-1 or Latin-9) character sets. ISO-8859-15 is more or less identical to ISO-8859-1 except that the OE/oe dipthongs are supported, replacing the seldom-used 1/2 and 1/4 glyphs. See also **luit(1) charsets(7) iso_8859-1(7)** and **iso_8859-15(7)** for more information.

LOCALE

An important means of ensuring the system knows to use UTF-8 is to make sure the locale is set appropriately. This is valid only when **Dumb Frotz** runs under Unix-ish systems.

Using the command **locale** will tell you what is currently in use. Using **locale -a**

will show you what's available. Then set your LANG evironmental variable to something appropriate by using one of these commands:

```
export LANG=C.UTF-8
export LANG=en_US.utf8
```

This can be put in your shell configuration file, be it **.profile, .bash_profile, .login, .bashrc,** or whatever.

It can also be set system-wide in the equivalent files in **/etc.**

SEE ALSO

ash(1) bash(1) csh(1) ksh(1) sh(1) zsh(1)

ENVIRONMENT

If the ZCODE_PATH environmental variable is defined, frotz will search that path for game files. If that doesn't exist, INFOCOM_PATH will be searched.

For the Alt key to be read correctly in an Xterm, the following lines should be in your .Xresources file:

```
XTerm*metaSendsEscape: true
XTerm*eightBitInput: false
```

FURTHER INFORMATION

The **Frotz** homepage is at https://661.org/proj/if/frotz/.

A **git(1)** repository of all versions of Unix Frotz back to 2.32 is available for public perusal here: https://gitlab.com/DavidGriffith/frotz/.

The bleeding edge of Frotz development may be followed there.

The Interactive Fiction Archive is a good place to find games to play with Frotz. Various ports and builds for Frotz may also be found here. Here is its URL:
http://www.ifarchive.org/

Most distributions of Linux and BSD include **Frotz** in their package repositories.
It is distributed under the GNU General Public License version 2 or (at your option) any later version.
https://www.gnu.org/licenses/gpl-2.0.en.html
This software is offered as-is with no warranty or liability. If you find a bug or would like **Frotz** to do something it doesn't currently do, please visit the above Gitlab website and report your concerns.

CAVEATS

The Z Machine itself has trouble with the concept of resizing a terminal. It assumes that once the text height and width are set, they will never change; even across saves. This made sense when 24x80 terminals were the norm and graphical user interfaces were mostly unknown. I'm fairly sure there's a way around this problem, but for now, don't resize an xterm in which frotz is running. Also, you should try to make sure the terminal on which you restore a saved game has the same dimensions as the one on which you saved the game.

Audio latency might be unreasonably long depending on the settings of your operating system. Linux generally has things right. The BSDs may need some sysctl(8) settings adjusted. See the sound(4) or audio(4) manpages for more information.

You can use a path like "/usr/local/games/zcode:$HOME/zcode" with $ZCODE_PATH or $INFOCOM_PATH because the shell will digest that $HOME variable for you before setting $ZCODE_PATH. While processing frotz.conf and $HOME/.frotzrc, a shell is not used. Therefore you cannot use environmental variables in the "zcodepath" option within the config files.

This manpage is not intended to tell users HOW to play interactive fiction. Refer to the file HOW_TO_PLAY included in the Unix Frotz documentation or visit one of the following sites:
http://www.microheaven.com/ifguide/
http://www.brasslantern.org/beginners/
http://www.musicwords.net/if/how_to_play.htm
http://ifarchive.org/

BUGS

This program has no bugs. no bugs. no bugs. no *WHAP* thank you. If you find one, please report it to the Gitlab site referenced above in **FURTHER INFORMATION.**

AUTHORS

Frotz was written by Stefan Jokisch for MSDOS in 1995-7.
The Unix port was done by Galen Hazelwood.
The Unix port is currently maintained by David Griffith <dave@661.org>.

CONTRIBUTORS

In 2019, a Kickstarter campaign was run to raise funds to pay Mark McCurry to overhaul the audio subsystem for the curses port of **Frotz.** The following people contributed $100 towards that effort:
Simon Martin
Dan Sanderson
Justin de Vesine
Daniel Sharpe

SEE ALSO

sfrotz(6) **dfrotz**(6) **nitfol**(6) **rezrov**(6) **jzip**(6) **xzip**(6) **inform**(1)