NAME

pinfo − play Infocom Inc. adventures

SYNOPSIS

pinfo [**−aAehoOpstvV**] [**−c** *context*] [**−i** *indent*] [**−l** *lines*] [**−m** *margin*] [**−r** *savefile*] [*display-flags*] [*datafile*[*.dat*]]

DESCRIPTION

This program allows most of the popular Infocom text adventures to be played on your system. The games must be Standard Series for the interpreter to function correctly; see the file *INFOTBL.TXT* in the product release for full details on all Infocom games.

The **−h** option will tell you what version of Z-Code Interpreter you need for that adventure. The informational printing options (**−h**, **−o**, **−O**, **−v**, and **−V**) will work with any adventure, but you won't be able to actually play any games except for Standard Series adventures.

OPTIONS

The command-line arguments allow for screen customizations, and also various forms of cheating and/or debugging.

Informational

These option will print information on the datafile to the screen. If any of these are given the game is not actually played, information is just printed.

Note that all these options except **−h** and **−V** will print information that you are not supposed to know: examination of this information on a game you haven't solved could seriously detract from your enjoyment of the game.

−h print the datafile header

−o print the object names, attributes and links

−O print the object tree

−v print the game vocabulary

−V verify the game data file to ensure it hasn't been corrupted (to get this same feature while playing use the **$verify** command)

Game Play

These options affect different aspects of game play. Some of them may definitely be considered cheating as they print information about the internals of the game as you are playing it.

−a display modifications to objects' attributes while playing

−A display value tests of objects' attributes while playing

−c *context*
set the number of lines of context to keep at the top of the screen when paging long output (default is 2)

−e echo each command before executing

−i *indent*
set the number of spaces in the left margin (default is 0). Note that only game output text will be indented; input prompts and input will not.

−m *margin*
set the number of spaces in the right margin (default is 2)

−p don't page long output

−P set the alternate prompt flag (this flag is not used in all games)

–r *savefile*
causes the interpreter to restore the saved game *savefile* after starting up.

–s don't print status line

–t display modifications of the object tree while playing

–T set the Tandy Licensing flag (this flag is not used in all games). On some games this causes printed text to be slightly different.

Interpreter Escape

Some of the above flags and options may also be modified during game play via interpreter escapes. Any command beginning with the escape character (normally "@") is assumed to be an interpreter escape command. The current commands and their values may be listed by giving just the escape character by itself. Current commands include:

attr Toggles the attribute printing variable (see **–a** above)

echo Toggles input line echoing (see **–e** above)

pager Toggles paged output mode (see **–p** above)

prompt
Toggles the alternate prompt (see **–P** above)

status Toggles status line printing (see **–s** above)

tandy Toggles the Tandy license flag (see **–T** above)

xfers Toggles the object transfer flag (see **–t** above)

Additional escape commands may be defined by individual terminal interfaces; see below.

TERMCAP

This section gives information specific to the **termcap** terminal interface. This interface is for use with UNIX machines or others with UNIX-compatible termcap or terminfo libraries. In addition to termcap and terminfo it also supports the termio, termios, and sgtty line-discipline libraries.

Options

These options are in addition to the above, general options:

–C *file* Specifies a file of completion words. If not given no words are available at startup and any words defined during play will not be stored. Note pinfo must be compiled with GNU Readline support to use this option.

–H *file* specifies a GNU Readline library history file to be used. If not given, history is not stored in a file. Note pinfo must be compiled with GNU Readline support to use this option.

Subshells

The TERMCAP version of the interpreter can invoke commands in a subshell or an interactive subshell in the normal UNIX fashion: i.e., if a command's first character is an exclamation mark (bang) ("!") then the remainder of the command is passed verbatim to a subshell. If there is no text after the exclamation mark then an interactive subshell is invoked.

The value of the environment variable **SHELL** is used as the subshell, or */bin/sh* if **SHELL** doesn't exist or is empty.

See the *GNU Readline* section for an exception when using GNU Readline.

GNU Readline

The TERMCAP version of the interpreter contains optional support for the GNU Readline library. This library allows input line editing and history retrieval, including full incremental searching and **csh**-style command-line editing. By default the key bindings are those of GNU Emacs, but the library is fully flexible and can be configured to work in vi mode, or you can roll-your-own bindings.

Because **csh**-style editing uses an exclamation mark as the first character for history substitutions, in order to get a subshell command (see above) you should escape the exclamation mark with a backslash (“\”) when using the GNU Readline library. For example:

```
>\!ls -l
```

In addition to the standard GNU Readline bindings, the single character command **?** will cause the interpreter to print the current history list to the screen.

See the GNU Readline manual for full details.

Completion

If the GNU Readline library is being used, then completion is available by pressing the TAB key. A list of all possible completions is available by pressing TAB twice in a row.

During normal game play the interpreter maintains a list of possible words to complete on. This list is read in from a file specified with **–C** (above) at startup and written out to that file at the end of the game. Modifications to the list are possible via special interpreter escape commands (see below).

Filename completion is available during save/restore processing, whenever the interpreter is asking for a filename.

Interpreter Escapes

Two new escape commands are defined in addition to the normal escapes if pinfo is compiled to use the GNU Readline library. These allow addition and deletion of words in the command completion list.

add *<word>*
: Add *<word>* to the completion list. Duplicates are not allowed.

del *<word>*
: Delete *<word>* from the completion list.

Environment Variables

When using the interpreter with TERMCAP support, the following environment variables are used:

SHELL
: If set, contains the full path of the users shell. This program is invoked if the shell command is typed at the game prompt with no arguments.

TERM
: This variable contains the terminal type used to display the output.

PI_FLAGS
: This variable should contain flags which you would always like present on the pinfo command line. Any flags given here will be overridden by flags actually on the command line, and be the next two variables in the case of **–H** and **–C.**

PI_HIST
: Contains the filename of the pinfo history file used with GNU Readline support (see above). Setting **–H** on the command line overrides the value of this variable.

PI_WORD
: Contains the filename of the pinfo command-completion word file used with GNU Readline support (see above). Setting **–C** on the command line overrides the value of this variable.

STREAM

There are no stream terminal interface specific flags or capabilitites.

DATAFILES

Infocom games consist of a datafile and an interpreter which executes the instructions contained in the data file to run the game. This program is essentially a portable datafile interpreter.

If you have the ‘‘Lost Treasures of Infocom’’ package for the IBM or Amiga you should have the datafiles readily available on the disks. If you have any Infocom game for the Macintosh then when transferring the program to your UNIX host be sure to **not** send the Macintosh resource sections (i.e., **don’t** enable the MacBinary option). Mac Infocom programs sans their resources are the datafiles needed for use with this program.

BUGS

Some of the data files have errors in their vocabulary lists. If the **−v** option detects such an error it will put square brackets (**[]**) around the offending word. Such a word may not be used in the game, since the interpreter will not recognize it. It actually can't be used with *any* interpreter, even the ones from Infocom itself; these are bugs in the data file.

AUTHORS

The original source code for this interpreter was the InfoTaskForce
: interpreter, version REV_C.

Paul D. Smith <paul_smith@dg.com>
: Modified from ITF REV_C to the current version. Many enhancements: see *NEWS* in the source distribution for a complete list of user-visible modifications.

Olaf Barthel <olsen@sourcery.mxm.sub.org>
: Contributed the Amiga terminal interface.

Mike Threepoint <linhart@remus.rutgers.edu>
: Contributed the MS-DOS terminal interface.

COPYRIGHTS

This pinfo interpreter is distributed under the GNU General Public License (see the file COPYING in the source directory).

Infocom, Inc. has gone out of business, but the rights to their games have been sold (various times!) and are currently still owned. All Infocom games are still protected and as such may not be copied.

Some games released by Infocom have a very restrictive copyright which states that the game may not be played on any platform except that for which it was sold. Please refer to the copyright notice contained in your game to decide whether you will be violating copyright laws before using it with this interpreter.

Now that Lost Treasures is available there is no excuse for not purchasing the Infocom games yourself; at about $2.50 to $3.00 a game, including tax, shipping, handling, manual, maps, and hints [in LTOI I only] this is a bargain the likes of which you probably won't see again in your lifetime! :-).

Let's all support Mediagenic in their release of Lost Treasures I & II and maybe they'll release the rest of the lost Infocom games.