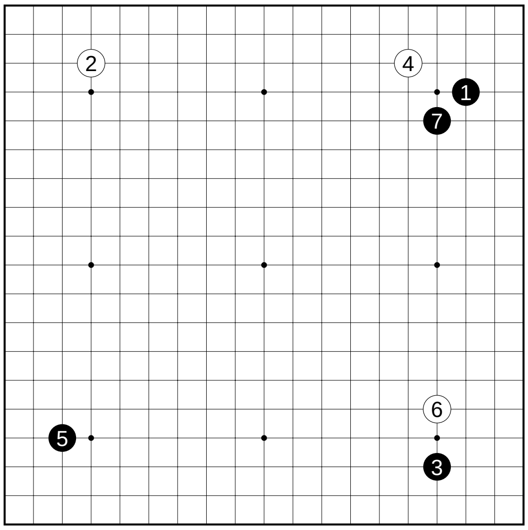


Liberty

The User's Manual

Jan van der Steen

July 11, 2001

All diagrams in the manual have been created with LIBERTY *2.0.*
All illustrations in the manual are screen images taken from LIBERTY *2.0.*
The manual has been typeset as òne LaTeX job.
The manual has been printed on an Océ 6750PS (508 dpi) laserprinter.

First printing, July 11, 2001

1 Introduction

Originally LIBERTY has been written for personal use only. I wanted to be able to write down *real time* Go analysis like professionals commenting on amateur games and the profound analysis of the *Shusaku* and *Go Seigen* games by Ronald Schlemper during those famous "Dutch Central Training" sessions in the 80's.

Later, during the years that I was the editor of the Dutch Go journal (*GO*), LIBERTY developed itself to a more general tool able to maintain large numbers of Go games and print quality diagrams for the magazine.

Again some time later, users started to interfere: "*Could you add this option...*", or "*How about that...*". In other words: things got completely out of hand.

At this moment LIBERTY has reached the second stage of development (out of three). There are still things to be done (especially the text and database interface) but in general the current group of LIBERTY users seems to be pleased with the LIBERTY 2.0 implementation.

However, if <u>you</u> have any complaints, suggestions or properties which you would like to see (changed) in the program please feel free to contact the author (see section 8).

2 About this manual

Actions within LIBERTY are controlled by either selecting options in the menu bar, pointing window attributes, hitting keys on the keyboard, or clicking the mouse inside one of the windows. I have decided to make these various means of input the main subjects of the manual. This will automatically provide a logical structure to this manual.

Section 3 will deal with the window attributes. What do they mean and how do you use them.

Section 4 will deal with the menu bar. The menu bar is the main control unit for LIBERTY. It enables the user to read and write games, to switch the *edit mode* and *edit attribute*, and eventually to leave the program.

Section 5 gives an overview of all the key bindings.

Section 6 deals with the mouse. The main application of the mouse is to enter edit attributes on the Go board. The shape of the mouse can change and this section explains why.

When you start up LIBERTY from a command line interpreter (a shell) you can supply the program with command line options. Besides, you can affect LIBERTY's behaviour by setting environment variables. Section 7 will describe them.

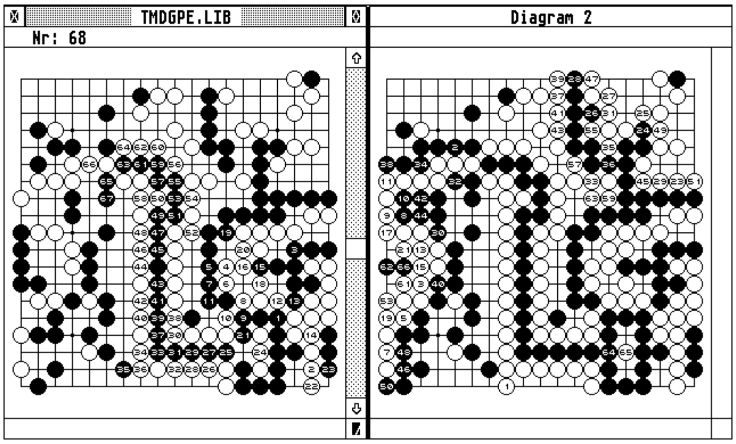
Section 8 finally, will give the aspects of LIBERTY which are subject to future changes/enhancements. If you are not satisfied with some aspect of LIBERTY then take a look at this section.

3 The windows

3.1 Introduction

A Go game is displayed on your screen in two windows: one for the game itself and one for variations on that game (called diagrams in this manual). You can distinguish the windows by their title bar. The game window displays the name of the current file or UNTITLED if the game has not been written to disk yet. The diagram window displays either DIAGRAM # ("#" is the number of the diagram currently displayed), or ANALYSE (if their is none).

In this section we will *not* explain how you can enter a Go game or how you can create diagrams. This will be explained in detail in the section dealing with the menubar (section 4). Instead, we will describe the functionality of the various window attributes like the slider, the arrow-, resize-, open- and close button, the scroll bar and the info bar:



The game and analysis window

3.2 The window attributes

Title bar

In the introduction of this section we already mentioned the filename or diagram number which is displayed in this window space. The color of the title bar (either gray or white) will indicate which window you are currently editing (the gray one). You can switch windows by clicking the left mouse button somewhere inside the window or by pressing either the $\boxed{\Leftarrow}$ or $\boxed{\Rightarrow}$ key.

Info bar

The info bar provides the user with various kind of information. You can toggle whether or not this information is displayed by pressing the i key.

> *NOTE* Sometimes an immediate message will be displayed in the info bar (when LIBERTY thinks it has to inform you about something). These immediate messages can not be suppressed.

The information supplied by LIBERTY:

dirty bit
: The first position of the info bar will contain a small circle (the dirty bit) if the current game has been modified since the last time you wrote the game to disk. If the dirty bit is displayed and you try to leave the program, LIBERTY will issue a warning to prevent that you will discard unsafed work.

edit attribute
: The next field in the info bar is used to display the current edit attribute. This can be either the current move number, color, mark or character depending on the current edit mode.

text...
: Following the edit attribute LIBERTY will display "(C)" if any text is attached to the current move. You can display text using the t key or TEXT....

edit mode
: The remaining portion of the info bar will display the current edit mode. This can be any of replay, insert or delete.

Arrow up

Go back one move.

Arrow down

Go forward one move.

Page up

Like some other buttons, the action triggered by pressing this button depends on the current window:

Game window
: Go back ten moves.

Diagram window
: Display previous diagram.

Page down

Depends on the current window:

Game window
: Go forward ten moves.

Diagram window
: Display next diagram.

Slider bar

The slider bar is used both to give the user some feedback about the current position within the game or a sequence of diagrams and to move quickly to a certain position within the game:

Game window

In the game window the position of the slider indicates the relative position in the game: All up, ie. touching the up-arrow button indicates the very beginning of the game. All down, ie. touching the down-arrow indicates the very end of the game. By dragging this slider (pressed left mouse button) you can quickly switch to other positions in the game.

Diagram window

In the analyse window the position of the slider indicates the number of diagrams attached to the current move: If the slider occupies the complete region between the up- and down arrow then there is just one diagram, if it occupies half this space, there are two diagrams (etc...). By dragging the slider you can switch from diagram conveniently.

Open box

Enlarge the current window to the full size of the screen.

Close box

Resize and/or reposition the windows to their default configuration.

4 The menu

The menu bar is positioned at the topmost part of the screen. Normally, the menu bar will be hidden. However, if you move the mouse near the top edge of the screen, it will be displayed and ready to use.

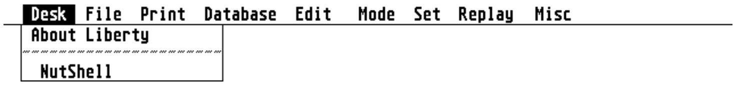
The menu bar of LIBERTY contains nine main titles:

```
 Desk  File  Print  Database  Edit  Mode  Set  Replay  Misc
```

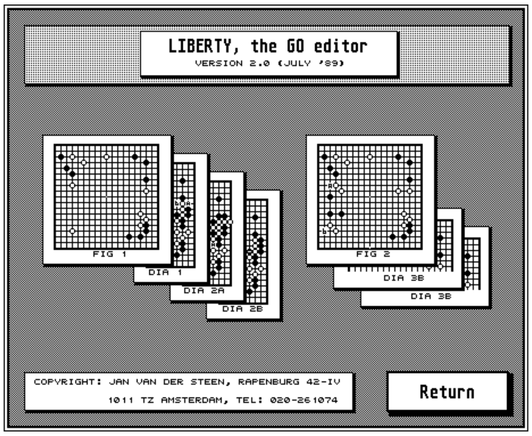
Each menu title has options associated with them which will all be explained in this section. The best thing for you to do now is to start up LIBERTY and to try out the options while they are explained to you.

4.1 Desk

Under DESK you'll find a ABOUT LIBERTY option and any desk accessory loaded before LIBERTY was started:



Selecting ABOUT LIBERTY will give you the copyright notice:

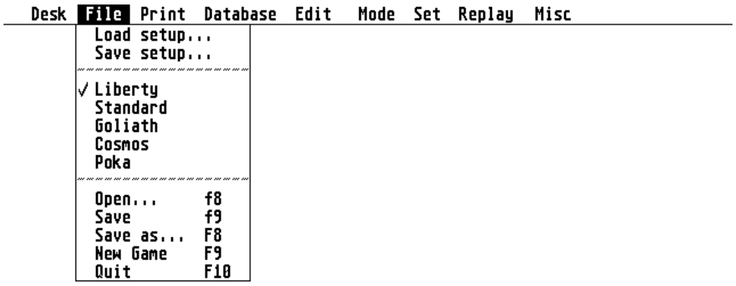


The copyright notice

You have a personal copy and are allowed to make copies for personal use. You are not allowed to distribute LIBERTY amongst other people.

4.2 File

The FILE entry contains the options associated with file input/output. By means of a gray colored separator the options have been divided into three sections:



These sections are:

- Reading and writing work environment(s)
- Switching the file format
- Reading and writing Go games

4.2.1 LOAD SETUP...

Liberty has a lot of options. The big advantage of many options is flexibility: every user can customise the program to his or her very special needs. To avoid that you have to specify those preferences over and over again, Liberty can write the complete setup to disk. With LOAD SETUP... you can read such a setup and they will take effect immediately.

When Liberty is invoked it will look for a file named LIBERTY.DFT in the current directory (or the directory set in the command line, see section 7) and, if it exists, load this file. So, if you want your settings to be loaded automatically you should save them as LIBERTY.DFT in the same directory where you stored the LIBERTY.RSC file.

> NOTE Items in the menubar which are followed by dots (...) indicate that a further dialog with the user is required. In this case the *file selector* will be presented to enable you to specify the name and folder of the setting file.

4.2.2 SAVE SETUP...

A complete list of the settings which are saved by Liberty can be found in appendix A. If you are interested in the contents of LIBERTY.DFT then you should use this menu item SAVE SETUP... and examine it. LIBERTY.DFT is an ASCII file so a text editor, the desktop *show* option or a program like *more* will do.

𝒩𝒪𝒯ℰ Each item in the file has a small comment and a listing of acceptable values making it more convenient to create or change the setup by means of a text editor.

4.2.3 LIBERTY

Currently, LIBERTY can read and write Go games in five different file formats:

Format	**Type**	**Saved attributes**
Liberty	binary	text, analysis, game
Standard	ASCII	analysis, game
Goliath	binary	game
Cosmos	ASCII	game
Poka	ASCII	game

The *Liberty* format (default) is a compressed and therefore binary format. The Go games will take up significantly less disk space than any of the ASCII formats. Besides, the *Liberty* format is the only format which will save all game attributes: the game itself, the analysis (markers, diagrams, ...) and the text. If you save a game with the name SHUSAKU.LIB (or SHUSAKU.XXX or just SHUSAKU), LIBERTY will also create the file SHUSAKU.TXT. The latter is an ASCII file containing general game information (players, date, ...) and any text comments written on the game.

𝒩𝒪𝒯ℰ Separate files for game data and text makes automatic post-processing more convenient.

4.2.4 STANDARD

Since there are as many formats to store Go games as there are programs to enter them, I added a file format which is human readable while still providing enough flexibility to be usable by LIBERTY. This intermediate format can be used to convert LIBERTY format to any other electronic storage format and visa versa. In appendix B you will find a description of the format.

𝒩𝒪𝒯ℰ The international discussion about a standard for electronic storage of Go games which has been started in September 1989 by Ishii Press has resulted in yet another format: the *Go World on disk* format. Since this format is powerful enough to cover all of LIBERTY's editing facilities it will eventually be implemented, replacing the current *standard* format.

4.2.5 GOLIATH

The last three formats are used by Computer Go programs. These formats are not necessarily available in later versions of LIBERTY, but for the time being they ensure the possibility to exchange Go games conveniently.

The first format is *Goliath* the program written by Mark Boon (Holland). Goliath has been World Champion Computer Go in 1989 and 1990.

4.2.6 COSMOS

Cosmos has been written by David Fotland (USA). This program ended 8th in the 1988 WK Computer Go. People who have access to internet, or some other computer network might know David by his regular publications on that medium.

4.2.7 POKA

Poka has been written by Harold Landman (USA). Harold is an enthusiastic Go programmer with some very interesting ideas indeed. His program *Poka* for example is based on a database of over 2800 Go games played by professionals (of which about 250 games have been entered by LIBERTY incidentally). So far, Landman has not been very successful in the World Championships but that is not the main issue of his work I guess.

In future, some of the latter three formats might become obsolete if the authors decide to conform to the Ishii format. Therefore, I would like to advice you to use the *Liberty* format to store your games. This is the only format which will surely be supported in the future.

4.2.8 OPEN... `f8`

Read a game from disk in the currently selected format.

> *NOTE* You should realize that a file *written* in a certain format should also be *read* in the same format. No harm is done if you accidentally read a file in a different format than it has been written in. LIBERTY will issue one (or more) error messages though.

> *NOTE* Like most other frequently used items in the menubar this item has a keyboard equivalent: `f8` (function key f8).

4.2.9 SAVE `f9`

Write the game to disk. If the game has no name yet, or if you want to write the game in another format than it has been read you should use SAVE AS....

> *NOTE* You should save frequently, especially if you are a novice. It's no fun to loose more than a quarter of an hour of work. Especially since hitting `f9` is all there is to write the game.

4.2.10 SAVE AS... F8

Used to write a copy of the current game using another filename, or to save a game which has not been written to disk yet, or to write a game in a different file format from the original.

> *NOTE* The keyboard equivalent is F8 which means pressing the shift and f8 key simultaneously.

4.2.11 NEW GAME F9

LIBERTY will clean up and present the user with an empty board. LIBERTY will never allow the user to throw away unsaved work however. If you accidently do try to throw away work you will get a dialog like:

Unsaved work warning

The arrows indicate that you have changed the game and the text. The edit part (like diagrams, marks in stones and characters on the board) has not been changed since the last save in this example.

4.2.12 QUIT F10

To leave the program. You can use function key f10 to save and quit.

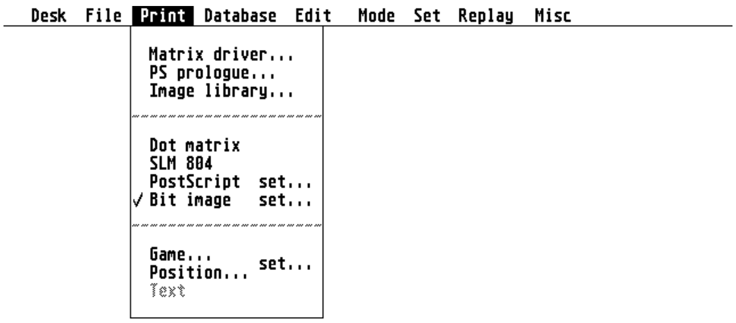
4.3 Print

The user can select between four different kinds of print output. Two of them produce immediate hardcopy output: the dot matrix and SLM 804 mode. The other two, POSTSCRIPT and bit image mode, create files which can either be spooled to a printer at a later stage (the *plain* POSTSCRIPT mode and the POSTSCRIPT bit image mode), or imported into other programs (the *encapsulated* POSTSCRIPT mode and the POSTSCRIPT bit image mode). All other bit image formats will enable you to convert the high resolution images to formats which are not directly supported by LIBERTY.

> *NOTE* Because of the large variety of printing devices and driver languages (POSTSCRIPT, PCL, HPGL, Epson-FX, ...) implementing a generic driver seems almost impossible. Yet, LIBERTY has tried to meet the printing facilities of most users by carefully taking into account contemporary printing and conversion tools wherever possible.

Once the format has been selected, LIBERTY enables the user to print the game (automatic or interactive) or just a position (which is not necessarily the same (see GAME and POSITION).

The print menu:



4.3.1 MATRIX DRIVER...

If your printer is not an Epson-FX compatible then you have to write your own printer driver. Please read the instructions in the file SAMPLE.HEX in the DRIVERS folder. With MATRIX DRIVER... you can load this driver and override the build in Epson-FX compatible codes.

4.3.2 PS PROLOGUE...

The file PROLOGUE.PS in the DRIVERS folder contains POSTSCRIPT procedures used by LIBERTY to print Go diagrams on POSTSCRIPT devices. If LIBERTY needs to know where you stored this file on disk it will tell you so.

4.3.3 IMAGE LIBRARY...

The file GOFONT.BIN (in the DRIVERS folder) contains the Go font. The Go font is a set of 77 bitmaps used to create a 6.8 x 6.8 cm (at 300 dpi) bitmap representation of Go diagrams. The font is used in SLM 804 and all bitmap modes. If LIBERTY needs the Go font it will tell you so.

The source of the Go font is plain ASCII thus enabling you to customize it. The format of this file and the tools to compile and preview the font are described in appendix C.

4.3.4 DOT MATRIX

Select this option if you intend to print on a matrix printer.

In *Dot matrix* mode you can print both text and Go diagrams. LIBERTY assumes that the matrix printer is an Epson-FX compatible. If your printer is not please refer to MATRIX DRIVER....

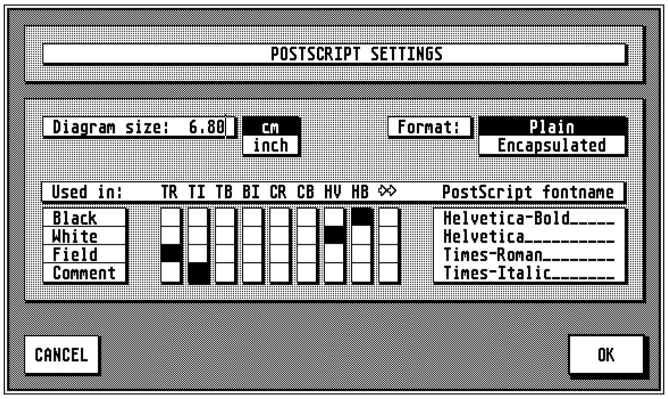
4.3.5 SLM 804

Select this mode if you want to print on the Atari laserprinter. LIBERTY will print six diagrams on every page (two columns of three diagrams).

4.3.6 POSTSCRIPT SET...

Select this mode if you want to produce a POSTSCRIPT file. At a later stage the file can be spooled to a POSTSCRIPT printing device. The POSTSCRIPT produced by LIBERTY is highly portable. It has been tested on a large variety of printers with various resolutions (including a 2400 dpi POSTSCRIPT typesetter).

With the associated SET... dialog you can specify further details:



Setting the postscript options

First, you should decide whether you want to create *plain* or *encapsulated* POSTSCRIPT. The main difference between both is that the first mode will create one file containing all diagrams and the latter will produce as many files as you print diagrams. The *encapsulated* mode also writes some additional information about fonts used and the size of the diagram enabling other applications to incorporate the file in their own environment (see appendix F for an example).

Next, you should set the diagram size. The size is that of the board not including stones or characters on the border. You can specify the size in centimeters or inches.

Finally, you can set the font to be used for numbers in both the black and white stones, the font for characters on the board and the font for the diagram caption. LIBERTY has buttons for the eight fonts available in almost all POSTSCRIPT devices. If you want another font then you can type it in the last column by selecting the [▷▷] button.

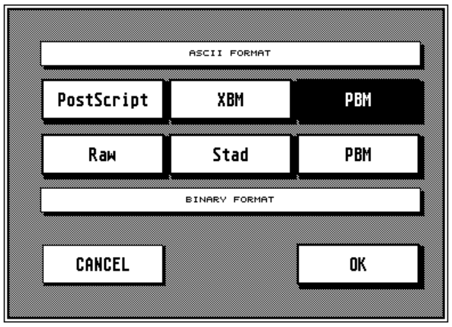
My experience is that the Helvetica font for numbers is a sound choice considering both legibility and availability. The decision whether the font face in the black stones should be bold depends on your printer. If it is a *write-black* printer then you should probably select the bold face to compensate for the heavy blackening.

NOTE Remember the [SAVE SETUP...] option in the [FILE] menu! This will also write your selections made in the POSTSCRIPT dialog.

4.3.7 [BIT IMAGE] [SET...]

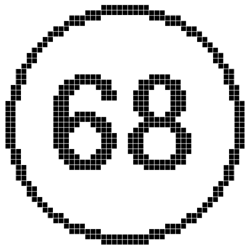
Select this option if you want to print a diagram as a bit image. The bit images created by LIBERTY will print 6.8 x 6.8 cm diagrams at a resolution 300 dpi.

LIBERTY supports six bit image formats, three of them are ASCII format and three are compressed format. The latter taking up much less disk space. Use the [SET...] menu option to select the desired format:

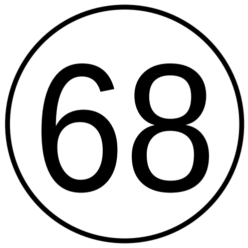


Setting the bitmap format

As a comparison in quality between POSTSCRIPT and bit image mode we will now do two small tests. First we take a closer look at the created image of a Go stone by magnifying it about 16 times:



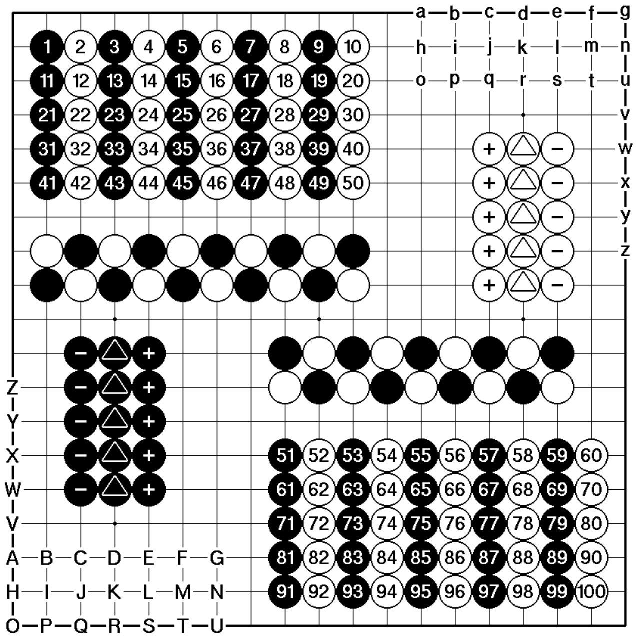
Bit image mode



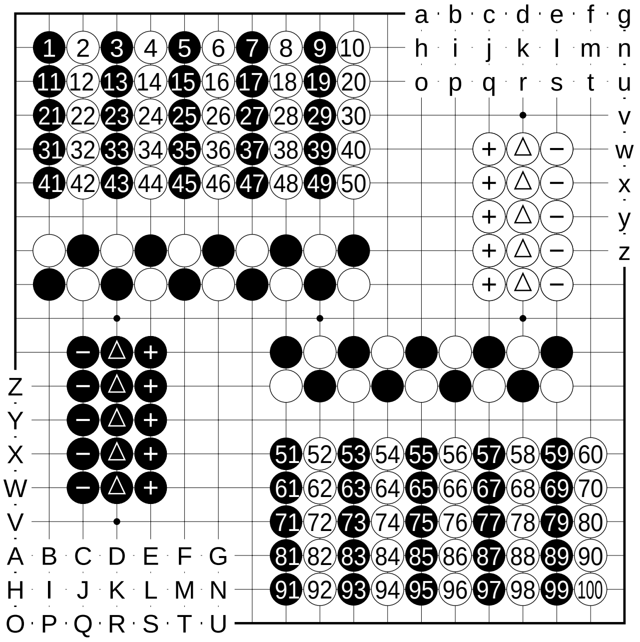
POSTSCRIPT *mode*

We can clearly recognize pixels in the bit image mode. (See also appendix C, section C.1.2. You will recognize the shape of digit "8".) In POSTSCRIPT mode however, curves are still smooth at this size. The reason being that in POSTSCRIPT mode a Go stone is designed using <u>curves</u> rather than <u>pixels</u> making the shape independent of the resolution.

The second test will show you that things are not as bad as they seem to be. When bit images are printed at the size they were designed for (see appendix D, the main differences in quality will almost completely vanish.



Bit image



PostScript

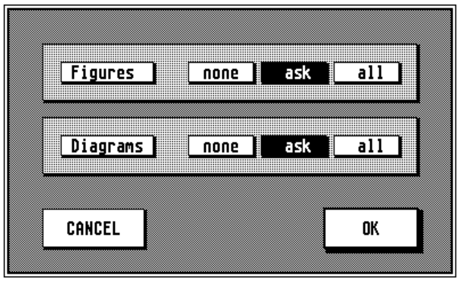
4.3.8 GAME...

This item will print the current Go game, ie. a diagram for every CONTINUE and DIAGRAM command given when you entered the game.

If their are more moves on the same spot in a diagram (a ko for example) Liberty will only print the first one and (in PostScript mode only) mention the others in the caption of the diagram (eg. *65 on 47, 68 on 22, 71 on 47*).

While Liberty is printing a game it will display the current diagram on the screen (thereby seemingly speeding up the process!).

With the SET... menu entry you can select whether you want to print automatically (ALL), interactively (ASK) or none (NONE). This can be set for both diagrams and figures:



Setting the printing mode

If Liberty needs a filename for output (in PostScript and bit image mode) it will pop up the file selector during printing.

4.3.9 POSITION. . .

LIBERTY will print an exact copy of the position on the screen.

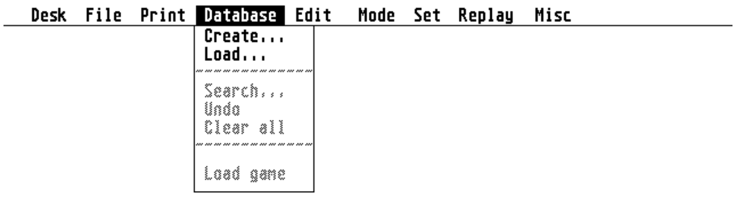
4.3.10 TEXT

Only selectable in DOT MATRIX mode. By adjusting SAMPLE.HEX (in the DRIVERS folder) you can customise the text output to your taste.

> *NOTE* A likely future development for LIBERTY will be to provide an automated LaTeX interface. LIBERTY will print both Go diagrams and text as a fully prepared LaTeX source ready to be typeset and printed. A sample of both source and output in such a setup is shown in appendix F.

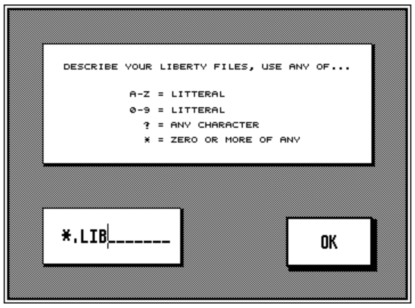
4.4 Database

Once LIBERTY is used regularly, you will notice the number of Go games increase rapidly. To provide a convenient access to these games a simple (but effective) database has been implemented in LIBERTY. In this section we will describe the possibilities. The database menu:



4.4.1 CREATE...

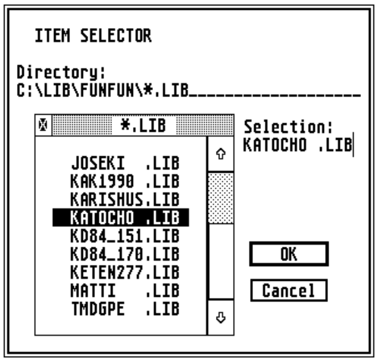
With this option you can create a new database. When you select this option LIBERTY will question you about several things. First of all it wants to know how it can recognise files containing Go games (in LIBERTY format). You can describe the filenames using the MSDOS wildcards ("*" and "?"):



File pattern for LIBERTY *games*

The "*" means zero or more of any character while "?" symbolises any character. The author always writes Go games as SOMENAME.LIB so he enters *.LIB as the file pattern to look for.

Next LIBERTY will pop up the file selector:



Folder and name for the database

In the *directory* field you should enter the folder where the Go games are stored. LIBERTY will store all the games found in here and all the folders found below this folder.

In the *selection* field you enter a filename for the database. The file will be written in the folder specified in the upper field.

At this stage you can still abort the whole process by applying the CANCEL button. Once you press the OK button LIBERTY will start creating the database.

When LIBERTY is finished it will give a full report of its work. A typical example of such a report:

CREATE... *report*

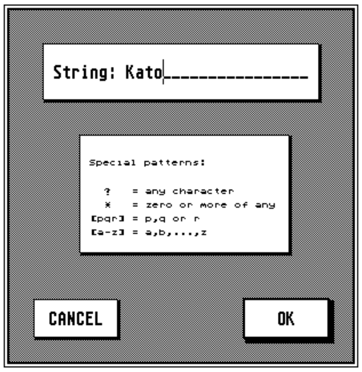
The created database file contains pure ASCII data, so you can examine it with an ordinary text editor.

4.4.2 LOAD...

To check this option out you can load the database of the Shusaku or Go Seigen games which are stored on the LIBERTY game disk. In the info line LIBERTY will report how many games are stored in the database.

4.4.3 SEARCH...

Once a database is loaded you can select games by string search:



Database search interface

The search algorithm used is case sensitive, so an "A" is different from an "a".

LIBERTY will only examine the games selected by the last search. If no search has been performed yet, all the games are searched. LIBERTY will report the number of hits in the info line. If a search was successful, the next three options will be enabled:

4.4.4 UNDO

This will undo the effects of the last search. LIBERTY has as many undo levels as you have given SEARCH... commands.

4.4.5 CLEAR ALL

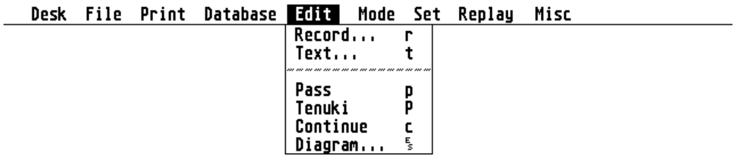
Will undo the effect of all SEARCH... commands. After a CLEAR ALL all the games in the database are again selected.

4.4.6 LOAD GAME

Will select the first game in the selected range. Each consecutive LOAD GAME will select the next game....

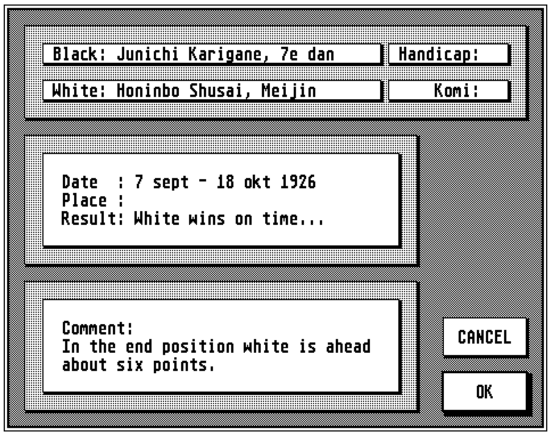
4.5 Edit

The edit menu contains the options to enter text, name of the players, make a diagram, create a figure and to enter *pass* and *tenuki* moves:



4.5.1 RECORD... r

Selecting this dialog will enable you to enter the name of the players, date, komi and more:

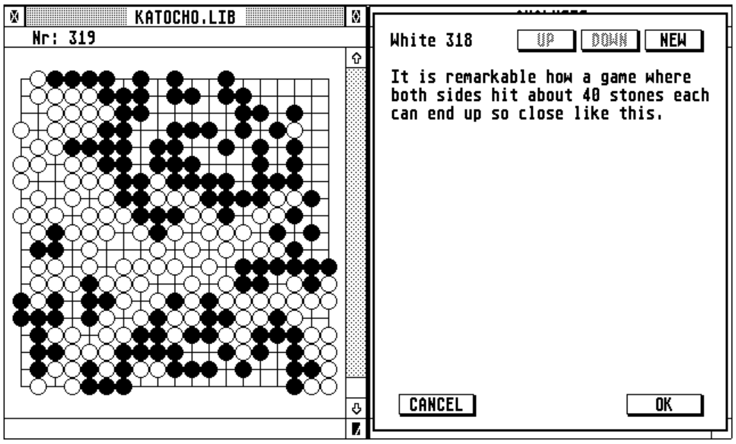


The game data

The user should be aware of the fact that these very data will be stored in the database. So keywords which you want to use to identify games should be entered in the record. For example, you might consider to enter the *fuseki* type (*san-rensei*, *chinese*, *shusaku*, ...) or the *joseki*'s occurring in the game (*nadare*, *magic sword*, *taisha*, ...).

4.5.2 TEXT... t

To enter text comment on the current move (in the game) or diagram:



The commentary dialog

There is no restriction (except memory) on the number of pages of text attached to the same move or diagram. The buttons to create extra sheets and to scroll up and down:

NEW

The current sheet is saved and an empty sheet is appended. This option will be disabled if the current edit mode is REPLAY, or DELETE.

UP

Display the previous sheet attached to the same move or diagram.

DOWN

Display the next sheet attached to the same move or diagram.

The edit rules for the text sheet are almost the same as other GEM edit objects:

`esc`	erase current line
`backspace`	erase previous character
`delete`	erase current character
`return`	goto next line

You can move around with the arrow keys.

NOTE If you have text attached to a move or diagram LIBERTY will indicate this in the info line with a “(c)”.

The format which LIBERTY uses to read and write text:

#123
This is comment on game move 123.
...
#
This text is also comment on move 123 since a move and/or diagram number was missing in the previous line.
...
#123 3
This is comment on the 3th diagram, attached to game move 123.
...

4.5.3 PASS `p`

To enter a pass in the game or diagram.

4.5.4 TENUKI `P`

To enter a tenuki in a diagram (not allowed in a figure).

4.5.5 CONTINUE `c`

A CONTINUE will start a new figure (or diagram) giving a new diagram if you print out the game. Numbered stones are replaced by unnumbered ones and marks and characters are removed.

> *NOTE* The Dutch Go Magazine normally publishes Go games with 50 moves in one diagram, so (normally) every 50 moves we insert a CONTINUE.

4.5.6 DIAGRAM `esc`

A DIAGRAM will create a diagram based on either the current figure or diagram position depending on the window currently topped. The diagram window will be automatically topped, the edit mode will be switched to INSERT and the edit attribute to NUMBERED. The keyboard equivalents to create diagrams are `l_shift` `esc` (a diagram based on game position) or `r_shift` `esc` (a diagram based on diagram position).

4.6 Mode

LIBERTY distinguishes four edit modes:

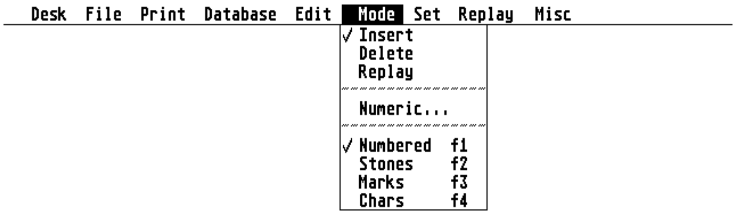
- INSERT
- DELETE
- REPLAY
- NUMERIC...

The first three modes toggle and use the mouse to edit the game. The last mode is a special mode using only the keyboard.

When LIBERTY is in INSERT mode, you can choose from any of four edit attributes:

- Moves (numbered)
- Stones
- Stone indicators (marks)
- Board indicators (characters)

Both the current edit mode and attribute are indicated in the menubar with a $\surd$ sign:



4.6.1 INSERT

The edit mode for most of the time. All edit actions are allowed (edit attributes, record, text and diagrams) except DELETE actions.

4.6.2 DELETE

Used to remove numbered stones, stones, marks and characters by pointing and pressing the mouse button.

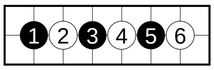
If you delete a marked stone, first the mark will be deleted and next the stone itself.

If you delete a move (a numbered stone) LIBERTY will automatically decrease moves with a higher numbers. This will be reported by LIBERTY in the info line of the currently active window.

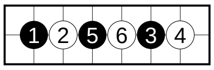
I want to illustrate the effect of decreasing numbers and advise you to try this example out for yourself so you can get acquainted with the effects of the DELETE command:

- Select NEW GAME
- Put LIBERTY in INSERT mode (in the MODE menu)
- Put LIBERTY in NUMBERED mode (in the MODE menu or `f1`)
- Enter some moves by pressing the left mouse button on different grid points

The position will look something like:



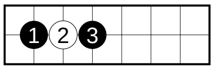
Now let's assume that move 3 and 4 were actually played after move 5 and 6, so we want to change the game into:



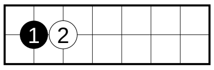
We put LIBERTY in DELETE mode and click on move 4. LIBERTY will now take three actions:

- It will rewind the game to move 4
- It will delete move 4
- It will decrease all move numbers higher than 4

All this will result in the next situation on the screen:



There are more moves but because the current move number is 4 (see the info line in the window) they are not visible on the screen. We also DELETE move 3 by clicking it:

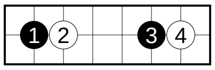


Go to the end of the game. You can do this by either:

- dragging the slider
- pressing ⇓ several times
- pressing `shift` ⇓ (ten moves at once)

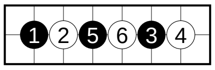
- pressing `shift` `clr home` (to end of game)

Now we see the following situation on the screen:



And indeed, LIBERTY has decreased move 5 and 6 two times (they have become move 3 and 4).

The final step is to enter move 5 and 6. We put LIBERTY in INSERT mode and click on the correct grid points. This will achieve the desired situation:



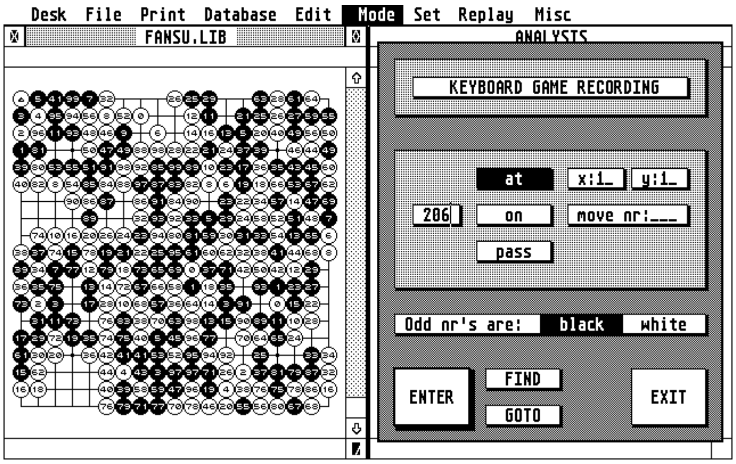
4.6.3 REPLAY

All edit actions are prohibited. The game is *read only*. An ideal way to browse through a game without making unintended changes.

4.6.4 NUMERIC...

Did you ever try to replay a game from a record sheet with 250 moves in it (Kido yearbooks for example). It takes a looong... time to find the next move. To make it easier to enter those games (and in the process making it more fun to replay it) LIBERTY has implemented the NUMERIC... mode.

Select NUMERIC... and ignore the warning. On the left side of the screen LIBERTY will display a Go board with a cursor (△) on the upper left gridpoint. On the right side of the screen LIBERTY will display the NUMERIC... dialog:



Entering a game with the keyboard

The basic idea of the NUMERIC... mode is that you enter the Go game row by row instead of sequentially. For each gridpoint you enter the move number in the field located about in the middle of the dialog box ("206") and next hit the `return` key (or press the ENTER button). The move will be displayed on the board and the cursor will advance to the next gridpoint. This process will continue until all moves have been entered. This is the default AT mode.

Special moves like ko's and passes can be entered using the ON and PASS modes. You can remove a stone at (x, y) by entering a "0" (zero) in AT mode at that point. You can find a move by entering the number and selecting the FIND button. You can go to (x, y) by entering the coordinates and selecting the GOTO button.

When you exit the NUMERIC... mode (with EXIT) LIBERTY will inform you about any missing moves. Apply the OK button for the next missing move (if any) or the CANCEL button to ignore them altogether and to quit NUMERIC... mode.

> NOTE Normally, the most efficient way to use the NUMERIC... mode is to first enter all visible moves by means of the keyboard and next to enter all invisible moves (ko, throw-in, ...) in the full screen editor using the mouse.

4.6.5 NUMBERED `f1`

This is the default edit attribute. For each click of the left mouse button on an empty grid point LIBERTY will enter a numbered stone on the board. After each move the color is toggled and the current move number is automatically increased. Use NUMBER... to change the current number.

If you enter a move with a number smaller than one of the moves already entered two things can happen: if the move number does not exist it will be inserted, if it *does* exist all move numbers will be increased to allow your move being inserted.

Since three digit numbers do not fit in the stones (on the screen) they are displayed modulo 100. So, for example, move "125" will be displayed as "25".

> NOTE While you enter moves LIBERTY will check if the move is legitimate (according to the Japanese rules). So, for example, suicide is not allowed.

4.6.6 STONES `f2`

A stone is a move without a number, normally used to enter handicap stones or to create some initial position. In STONES mode LIBERTY will not toggle the color. Use COLOR... to change it.

> NOTE LIBERTY will not check if the moves are legitimate. So you can use the STONES edit attribute to create "funny" positions.

4.6.7 MARKS f3

You can use marks to indicate stones. Marks are only allowed in stones without a number, ie. entered in STONES mode or entered in NUMBERED mode after having applied a CONTINUE. Marks are normally used to indicate groups of stones (a weak group, influence, ...). Use MARK... to set the current mark.

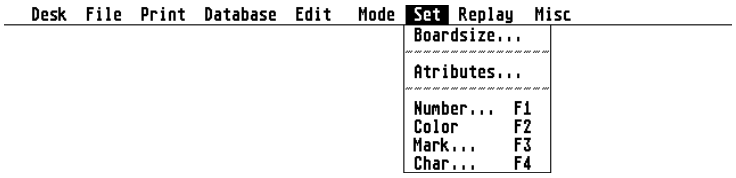
4.6.8 CHARACTERS f4

With the CHARACTERS attribute you can put characters on empty board points to indicate special points. Allowed range is the complete character set of the Atari, but when you print the game it is not assured that LIBERTY actually can print all of these characters. LIBERTY only supports a, b, ..., z, A, B, ..., Z in all print modes.

The character will be reset to “A” after a DIAGRAM and CONTINUE command. Use CHAR... to set the current character.

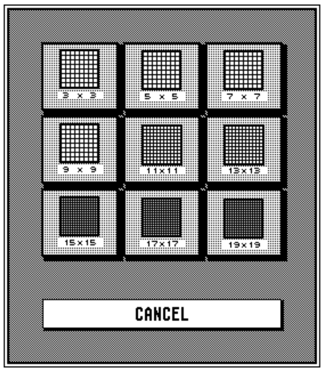
4.7 Set

You can set the boardsize, the current edit attribute and the graphic attributes which will be displayed on the screen:



4.7.1 BOARDSIZE

You can select any size in the range of 3, 5, . . . , 19:

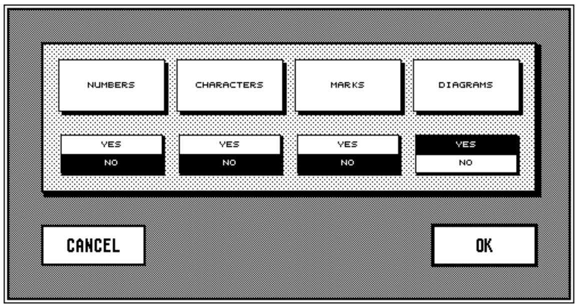


Changing the board size

Changing the boardsize will have the same implications as selecting NEW GAME. See that paragraph for the consequences.

4.7.2 ATTRIBUTES. . .

In REPLAY mode some people don't like numbers, marks and characters to be displayed. Also the automatic display of diagrams can be quit distracting. Therefore, LIBERTY has the possibility to suppress any of them:

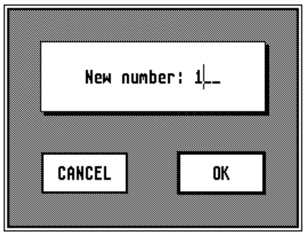


Affecting the graphics

Default LIBERTY will display all attributes.

4.7.3 NUMBER... F1

You can set the number in the range: 1, 2, ..., 999:



Setting the move number

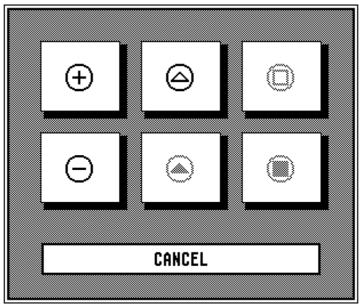
LIBERTY will switch automatically to the appropriate position.

4.7.4 COLOR F2

To toggle the color. So if the current color is black it will become white and visa versa.

4.7.5 MARK... F3

LIBERTY has three marks: +, – and △:

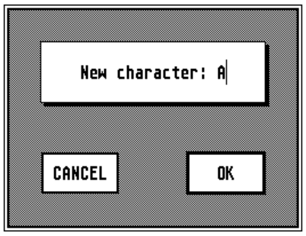


Setting the stone marker

The mouse will take the shape of the current mark.

4.7.6 CHARACTER... F4

Set the current character to be used on empty grid points:

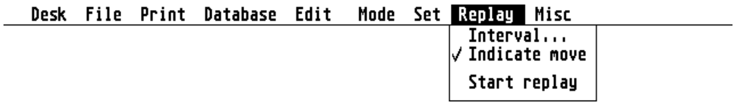


Setting the character

Allowed range is the complete Atari character set. However, only a–z, A–Z are guaranteed to be printable in <u>all</u> printing modes.

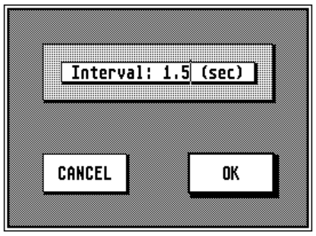
4.8 Replay

Once in a while it is nice just to sit down and watch a Go game being replayed. To make this possible, an automatic replay mode has been added to LIBERTY. The behaviour during replay can be influenced slightly by the user. The options:



4.8.1 INTERVAL...

To set the time delay between two consecutive moves:



Setting the time delay

Default is 1.5 seconds. LIBERTY will always pause for three seconds after each diagram and for five seconds at the end of the game before restarting it.

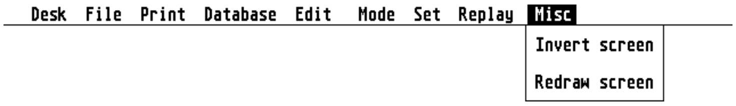
4.8.2 INDICATE MOVE

Default LIBERTY will display every move three times, making it easier to keep track of the game. With INDICATE MOVE you can toggle this behaviour. A $\surd$ sign in the menubar means this option is enabled.

4.8.3 START REPLAY

LIBERTY will start to replay the game automatically starting at the current position. You can pause/continue with `space` and exit with `return`.

4.9 Misc



4.9.1 INVERT SCREEN

The screen color will toggle white or black. By default LIBERTY will start up with a white screen.

4.9.2 REDRAW SCREEN

A reminiscent of a previous version of LIBERTY. Probably, you'll never need it.

5 The keyboard

Most of the keyboard options have already been mentioned in the menubar section. There is one key which deserves special attention: the `backspace` key. We will describe it in detail.

First an overview of all the key bindings of LIBERTY:

5.1 Reading and writing files

`f8`	Open...
`F8`	Save as...
`f9`	Save
`F9`	New game
`f10`	Save and quit
`F10`	Quit

5.2 Moving around

`help`	Indicate last move played
`g`	Go to move number...
$\boxed{\Downarrow}$	One move forward
$\boxed{\Uparrow}$	One move backward
`shift` $\boxed{\Downarrow}$	Ten moves forward
`shift` $\boxed{\Uparrow}$	Ten moves backward
`clr home`	Begin of game
`shift` `clr home`	End of game
`tab`	Go to (and show) next diagram
`ctrl` $\boxed{\Downarrow}$	Next diagram (in diagram window only)
`ctrl` $\boxed{\Uparrow}$	Previous diagram (in diagram window only)
$\boxed{\Leftarrow}$	Switch to figure
$\boxed{\Rightarrow}$	Switch to diagram

5.3 Editing

backspace	Remove last action (see 5.5)
undo	Recover from last backspace
r	Enter record...
t	Enter text...
p	Enter a pass
P	Enter a tenuki
c	Enter a continue
f1	Switch to mode numbered
f2	Switch to mode stone
f3	Switch to mode mark
f4	Switch to mode character
F1	Set number...
F2	Toggle color
F3	Set mark...
F4	Set character...
l_shift esc	Create diagram based on current figure
r_shift esc	Create diagram based on current diagram

5.4 Miscellaneous

i	Hide/show info line
ctrl m	Show free memory
ctrl M	Show largest free memory segment
ctrl u	Show the programs owner identity

5.5 The `backspace` key

In section 4.6.2 we described the DELETE mode of LIBERTY. The DELETE mode was used to delete attributes by pointing the mouse at them. But what about "*invisible*" actions like a pass, a continue, a diagram etc?

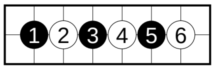
If the user hits the `backspace` key the last action will be removed. This can be any of:

- numbered stone
- pass (or tenuki)
- stone
- mark
- character
- continue
- diagram

If the target of the `backspace` is an "*invisible*" object like a pass or a diagram LIBERTY will ask for confirmation.

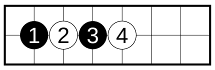
If the previous action was a numbered stone it will be removed. However, in contrast to the DELETE mode, moves with a higher number will not be shifted downward.

Let me explain this with a simple game:

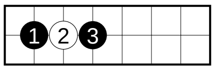


If we want to remove move 3 and 4 with the `backspace` then we should first go to the position where move 4 was the last move played. You can do so conveniently by double clicking the stone numbered 4.

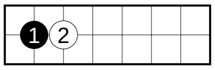
The screen will then show:



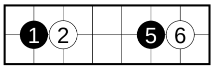
Now if you hit the `backspace` key:



And if you hit it again:



If you display the complete game on the screen (with `shift` `clr home` for example):



So, you have created two holes in the game: move 3 and 4 are missing.

You can fill in the hole by setting the move number to 3 (with NUMBER... in the set menu) and enter it. LIBERTY will report in the info line:

"Missing move inserted"

In the same way you can enter the missing 4th move.

6 The mouse

Only the left mouse button is used. Both the functionality and the shape of the mouse vary depending on the circumstances. This section will deal with both.

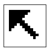
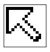
6.1 Using the mouse

The mouse is used for three purposes:

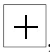
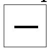
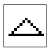
- To select GEM objects like menu items, dialog buttons and window attributes (sliders, arrows, ...).
- To edit the Go board. This can be either by entering edit attributes (numbered stones, unnumbered stones, marks and characters) or removing them (in DELETE mode).
- To go to a certain move. If you double click the mouse on a stone which was entered in numbered mode, LIBERTY will rewind the game to the position where it was the last move played.

6.2 The shape of the mouse

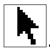
The shape of the mouse gives you feedback about the current edit mode and the current edit attribute:

NUMBERED
: The shape of the mouse is a fat arrow. The color of the mouse indicates the color of the next move,  (black) or  (white).

STONES
: Like NUMBERED.

MARKS
: The shape of the mouse indicates the current mark:
,  or .

CHARACTERS
: The shape of the mouse is: .

DELETE
: The default GEM shape of the mouse: .

7 Startup options

The user can affect the behaviour of LIBERTY during startup either by setting environment variables or by supplying command line options.

7.1 Command line options

If you invoke LIBERTY from a shell (command line interpreter) you can give it various options:

`-[12345]`
Selects the file format

`1` = *liberty*

`2` = *standard*

`3` = *goliath*

`4` = *cosmos*

`5` = *poka*

`-[gG] directory`
Selects the game directory

`-[dD] directory`
Selects the default directory (settings and resource file)

`-[lL] directory`
Selects the library directory (for drivers)

`file`
The game which LIBERTY has to read

7.2 Environment variables

If the environment variable GOGAMES is set then this will indicate the directory (folder) where LIBERTY will look for Go games. The value of GOGAMES will be overrided by the "`-[gG] directory`" command line option.

The directory specification may be any of the following forms:

- E:\LIB\SHUSAKU\
- E:\LIB\SHUSAKU
- \LIB\SHUSAKU

8 Future enhancements

As I already mentioned in the introduction of this manual some parts of LIBERTY still need improvement in my opinion.

First of all the text interface. It would be more pleasant to have a GEM window were the text could be displayed simultaneously. In the mean time the text editor's capabilities could then be extended.

Next, the database interface. The database should be displayed in a window. Then the user would be able to select a game by selecting it with the mouse. A search would highlight a selected part in the window, or would show only the selected games.

> *NOTE* Both this and the previous enhancement will probably come simultaneously, as their implementation share a considerable amount of code.

Finally, release 3.0 of LIBERTY will almost for sure provide an interface to TEX. It will be implemented as a print mode, producing a ready to typeset source file.

Of course, all reactions of users will be taken into account too in future releases. So, if you have any other suggestions than the ones listed above please let me know:

Jan van der Steen
Rapenburg 42–IV
1011 TZ Amsterdam
Holland
(+31) 020 – 626.10.74 (private)
(+31) 020 – 592.41.45 (work)
(+31) 020 – 592.41.99 (fax)

A The Liberty settings

This section will give an overview of the settings which are saved when you apply the SAVE SETUP... option. As you will notice when you study the listing, the format is not really designed to be edited by humans. When you want to make changes you are advised to do so from within Liberty.

```
#
# File information
#
 1# Default game folder  (pathname)     #A:\LIB
 2# File format          (0-4)          #0
#
# Patterns for various file formats
#
 3# Liberty  pattern     (*, ?)         #*.*
 4# Standard pattern     (*, ?)         #*.STD
 5# Goliath  pattern     (*, ?)         #*.GAM
 6# Cosmos   pattern     (*, ?)         #*.COS
 7# Poka     pattern     (*, ?)         #*.POK
#
# File suffixes for various file formats
#
 8# Liberty  file suffix (*, ?)         #LIB
 9# Standard file suffix (*, ?)         #STD
10# Goliath  file suffix (*, ?)         #GAM
11# Cosmos   file suffix (*, ?)         #COS
12# Poka     file suffix (*, ?)         #POK
#
# General print information
#
13# Output device        (0-3)          #3
14# game_range           (0-2)          #1
15# dia_range            (0-2)          #1
#
# PostScript printer information
#
16# PS prologue folder   (pathname)     #A:\LIB\DRIVERS
17# PS prologue name     (filename)     #A:\LIB\DRIVERS\PROLOGUE.PS
18# PS output            (pathname)     #A:
19# PS diagram size      (0-9999)       #680
20# PS diagram units     (0-1)          #0
21# PS output format     (0-1)          #0
22# Font in black stones (0-7)          #7
23# Font in white stones (0-7)          #6
24# Font on the board    (0-7)          #0
25# Font for commentary  (0-7)          #1
26# User black font      (string)       #Helvetica-Bold
27# User white font      (string)       #Helvetica
28# User field font      (string)       #Times-Roman
29# User comment font    (string)       #Times-Italic
#
# Matrix printer information
#
```

```
30# Matrix driver folder (pathname)     #A:\LIB\DRIVERS
31# Matrix driver name   (filename)     #
#
# Bit image output
#
32# Bit image format     (0-5)          #0
33# Image library folder (pathname)     #A:\LIB\DRIVERS
34# Image output folder  (pathname)     #A:
35# Image library name   (filename)     #A:\LIB\DRIVERS\GOFONT.BIN
#
# Edit settings
#
36# Boardsize            (1-9)          #9
37# Edit mode            (0-2)          #0
38# Edit attribute       (0-3)          #0
#
# Database information
#
39# Database folder      (pathname)     #A:\LIB
#
# Screen information
#
40# Show the numbers     (0-1)          #0
41# Show the characters  (0-1)          #0
42# Show the marks       (0-1)          #0
43# Show the diagrams    (0-1)          #0
44# Screen color         (0=W; 1=B)     #0
```

B The Standard format

B.1 Introduction

The standard format has been added to LIBERTY to make conversion to and from other electronic Go game formats easier.

B.2 The format

Empty lines and lines starting with a "#" sign are ignored. All other lines will contain the following information fields:

ACTION WINDOW COLOR NUMBER X Y

These items may be separated by spaces and/or tabs. Lines may not exceed 80 characters. All characters above 80 are silently truncated. Every line should contain six fields, even if not all of them are actually used. If a line doesn't contain six fields LIBERTY will complain and ignore it.
In the following table you will find an overview of the allowed ranges in various fields.

Legitimate ranges in Standard format

ACTION		WINDOW	COLOR	NUMBER	X	Y
0	CONTINUE	0–1	...	...	...	...
1	DIAGRAM	...	...	...	...	...
2	BOARDSIZE	...	...	1, 3, ..., 19	1–19	1–19
3	STONE	0–1	1–2	...	1–19	1–19
4	NUMBERED STONE	0–1	1–2	1–999	1–19	1–19
5	MARKED STONE	0–1	1–2	1, 2, or 3	1–19	1–19
6	CHARACTER	0–1	...	65–122	1–19	1–19
7	RESERVED	...	...	...	...	...
8	HIT STONE	...	...	...	1–19	1–19
9	PASS	0–1	1–2	...	...	...
10	TENUKI	0–1	1–2	...	...	...

We will discuss each field in detail.

B.2.1 ACTION

The ACTION represents the edit primitive. Allowed ACTION numbers are in the range 0, 1, ..., 10.

Most of the actions will be clear after having reading this manual except for code 8 (*hit stone*). The hit stone(s) comes after the move which has caused the hit. If the user doesn't feel to enter hit stones (or doesn't know about any hit stones), these data may be omitted provided there are no marks, stones, characters or diagrams. Entering and leaving NUMERIC... will build up the game including these hit stones.

B.2.2 WINDOW

WINDOW indicates whether an action refers to the game (0) or a diagram (1). Switching from game to diagram is done with a *diagram* ACTION. If the user has omitted this entry, LIBERTY will insert one.

B.2.3 COLOR

COLOR indicates the color of stones. Black is 1 and white is 2.

B.2.4 NUMBER

Used to give additional information if necessary. The way the information is interpreted depends on ACTION:

BOARDSIZE
: Acceptable boardsizes are 1, 3, ..., 19. BOARDSIZE is always an odd number. If the number is even it will be decreased to the nearest odd size available.

NUMBERED STONE
: The allowed move numbers are 0, 1, ..., 999. This has to do with internal limitations of LIBERTY.

MARKS
: The marks allowed are + (1), – (2) and $\triangle$ (3).

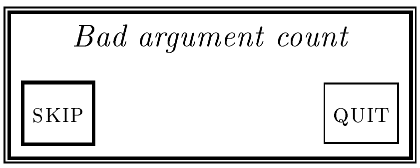
CHARACTERS
: The characters are interpreted as ASCII code (A=65, B=66, ..., Z=90; a=97, b=98, ..., z=122). LIBERTY can cope with: $32 \leq$ charcode ≤ 255, but most printers can only cope with A–Z, ..., a–z. Legal input therefore is: $65 \leq$ NUMBER ≤ 122.

B.2.5 X and Y

X and Y indicate the position on the board for the specified attribute. The upper left corner is $(1, 1)$. LIBERTY refuses input beyond the edges of a 19 x 19 board.

B.3 Error messages

While LIBERTY is reading a standard file, two error messages might be given:



This error is produced by an input line without exactly 6 arguments. The user can skip the line, or decide to abort the read process completely.

Variable var *out of range*

SKIP QUIT

This error message means *var* had an illegal value, ie. was out of the range listed in table B.2. Like the first error message, the user can skip the input line or quit.

C The Go font

When Liberty is printing on the SLM 804 or in bitmap mode it uses a set of glyphs to construct the bitmaps of the Go diagrams. These glyphs, implemented as 300 dpi bitmaps, describe the basic elements of a Go diagram. The complete set builds up the Go font. In this appendix the format of the font will be described enabling you to customize it to your own taste.

To speed up the process to read the Go font Liberty doesn't read the source GOFONT.BDF but the library GOFONT.BIN instead which you can compile from the source (C.2).

This library can also be used to preview the font on the Atari screen (C.4).

Since the library contains the same information as its original source you don't need to keep the latter but can extract it when necessary (C.3).

C.1 The Go font format

The format of the Go font strongly resembles the X11 BDF format (Bitmap Distributed Format). The main difference being that the character bitmaps are specified using a kind of WYSIWIG notation whereas the BDF format specifies them using hexadecimal ASCII codes.

C.1.1 The header

The header of the font contains global information about the font. The most important being the resolution of the intended output device (300 x 300 dpi), the maximum height of the characters (45 bits) and the number of characters in the font (77). You are strongly advised not to change this header:

```
STARTFONT 2.1
FONT GO_Toolkit
SIZE 45 300 300
STARTPROPERTIES 1
COPYRIGHT Liberty 1991, -- Go Diagram toolkit --
ENDPROPERTIES
CHARS 77
```

C.1.2 The character bitmaps

The shape of each character in the font is specified as a bitmap. As an example we will show the character "8", "+" and "a". To save space we print them next to each other. We have also decreased the line distance somewhat to (almost) equalize the distance between rows cq. columns.

```
STARTCHAR eight         STARTCHAR plus          STARTCHAR a
ENCODING 75             ENCODING 12             ENCODING 15
BBX 12 18 0 0           BBX 15 15 0 0           BBX 12 14 0 0
BITMAP                  BITMAP                  BITMAP
```

```
----####--------
--########------
--###--###------
-###----###-----
-###----###-----
-###----###-----
-###----###-----
--###--###------
---######-------
--########------
-###----###-----
###------###----
###------###----
###------###----
###------###----
-###----###-----
-##########-----
---######-------
ENDCHAR
```

```
------###-------
------###-------
------###-------
------###-------
------###-------
------###-------
###############-
###############-
###############-
------###-------
------###-------
------###-------
------###-------
------###-------
------###-------
ENDCHAR
```

```
---######-------
--########------
-###---####-----
-###----###-----
--------###-----
-----######-----
--#########-----
-#####--###-----
####----###-----
###-----###-----
###----####-----
####--#####-----
-###########----
--#####--###----
ENDCHAR
```

The character "–" denotes a white bit and "#" a black bit.

The characters have been padded on 8 bits but please note that the trailing white bits are not significant. For example, the four last columns of white bits in character "8" and "a" could have been omitted.

The information about the *width* and *height* of a character is extracted from the BBX line. This line should contain four integers which specify respectively the *width* and *height* of the character and the two integers normally used for the displacement in x and y of the bitmap with respect to the origin (but not used in LIBERTY).

When LIBERTY displays a character it will center it based on these *width* and *height* values. Numbers are treated slightly different. They are centered vertically based on the *height* of the largest character. The characters within a number are positioned 2 bits from each other and the resulting combination is centered horizontally (see the bitmap image of the Go stone in section 4.3.7).

C.2 The BDF2LIB program

During printing LIBERTY doesn't load the plain ASCII file GOFONT.BDF but the compiled library GOFONT.BIN instead which reads much faster.

You can compile a font library with the BDF2LIB program:

bdf2lib GOFONT.BDF GOFONT.BIN

bdf2lib will never overwrite existing files.

C.3 The LIB2BDF program

You can extract the font description file from the font library with LIB2BDF:

LIB2BDF GOFONT.BIN GOFONT.BDF

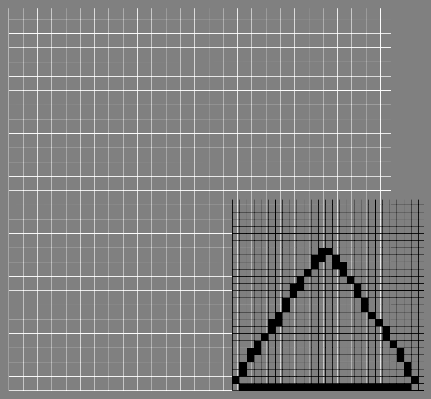
lib2bdf will never overwrite existing files.

C.4 The TSTFONT program

TSTFONT will display a sequence of Go diagrams on the screen using all of the characters in the font thus enabling you to evaluate the quality of the font. The usage of the program is:

TSTFONT GOFONT.BIN

During execution you can call the next diagram with `space` and quit with `q`.



The bit image *No problem* *Decision time* *Deformation*

An image sample (the 1^{st} image) is displayed at two resolutions: the resolution it was designed for (the 2^{nd} image) and at a slightly larger resolution (the 3^{rd} image). In the latter case the device driver is bound to make decisions whether or not bits should be turned on or off since some of them are partly filled. If we simplify this decision process (by consequently turning off partially filled pixels) we get the 4^{th} image. As can be seen from this example a mismatch in resolution will give deformation since image information is lost.

D.2 Avoiding deformation

As can be understood from the example above we can avoid deformation if we avoid that the output device has to make decisions about the pixels which have to be turned black. This is the case when the device resolution is a whole number $(1, 2, \ldots, n)$ times the bitmap resolution.

In section 4.3.7 we displayed a Go diagram at 8.08cm instead of its design size of 6.8cm. The reason can now easily be understood. The printer used to print this manual had a resolution of 508dpi (20 lines/mm actually). Displaying a 855x855 image at its natural size would have resulted in a 855/508 = 1.68 inch (about 4.27 cm) diagram. Considering that this would be rather small for the quality comparison purpose, we decided to magnify the image with 2 (a whole number thus avoiding deformation): 8.54cm, giving a board size of 8.54 x 18/19 = 8.08 cm.

E The convert tool

With **convert** you can translate Go games in LIBERTY format to either *cosmos*, *standard*, *goliath* or POSTSCRIPT format.

When producing POSTSCRIPT you can set the diagram breakpoints using commandline options thereby overriding the breakpoints stored in the LIBERTY file itself.

The options of **convert** are:

–l Produce POSTSCRIPT output (default). When you select this mode some additional options will be valid. They will be discussed below.

–p *file*
: Only in POSTSCRIPT mode. Specify *file* as the POSTSCRIPT prologue.

–w Only in POSTSCRIPT mode. Specifies that the target printer is a *write-white* engine. The font used in black stones will be the same as the one used in white stones.

–b Only in POSTSCRIPT mode. Specifies that the target printer is a *write-black* engine. The font used in black stones will be fatter compared to the font used in white stones.

–f *list*
: Only in POSTSCRIPT mode. The user specifies the list of figures to print. By specifying this list the user overrides the specifications in the input file. **Convert** recognizes two formats:

  %n Print a figure every n moves and one for any remaining moves.

  k+l+$\cdots$+n
  : Print a figure at move k, l, $\ldots$, n. A zero will skip the remaining moves.

  Some examples:

  –f *%50*
  : Print the game with figures of 50 moves each.

  –f *50+100+0*
  : Print two figures with move (1–50) and (51–100) and skip the remaining moves.

  –f *0*
  : Print no figures at all.

–d *list*
: Only in POSTSCRIPT mode. The user specifies a list of diagrams to be printed. **Convert** recognizes only one format:

  k+l+$\cdots$+n
  : Print the diagrams numbered k, l, $\ldots$, n.

  Some examples:

–d *1+3+6*
  Print diagram 1, 3 and 6 (if they exist).

–d *0*
  Print no diagrams at all.

–o *file*
  Write the output to *file* instead of standard output.

F The LaTeX publication environment

The columns below show the source (left) and output (right) of a typical Go publication sample. The main languages used are LaTeX for plain text and PostScript for the Go diagram specifications. The *epsf macros* which come with Tom Rokicki's **dvips** driver are used to merge these two languages.

```
\newcommand{\black}[1]{%
    {\bf Black #1: }%
}
\newcommand{\white}[1]{%
    {\bf White #1: }%
}

\begin{tabular}{l@{ : } l}
  {\bf Black}
    & {\em Iwamoto Kunwa, 8 dan}\\
  {\bf White}
    & {\em Go Seigen, 8 dan}    \\
  {\bf Place}
    & {Tokyo, Japan}            \\
  {\bf Date}
    & {7-9 July 1948}           \\
  {\bf Result}
    & {\em White by 1 point}
\end{tabular}
\vspace{.5cm}

This game was played during one of
the famous {\em chu-ban go}'s
sponsored by the {\em Mainichi}
newspaper.

\begin{center}
    \vspace{.5cm}
    \centerline{%
\epsfbox{diagrams/dia1.ps}%
    }
    \vspace{.3cm}
    {\em Figure 1: 1--7}
    \vspace{.5cm}
\end{center}

\white{4}
A most common {\em kakari}.

\black{7}
Typical {\em fuseki\/} in those
days: the {\em Shusaku fuseki}.
```

Black	: *Iwamoto Kunwa, 8 dan*
White	: *Go Seigen, 8 dan*
Place	: *Tokyo, Japan*
Date	: *7-9 Juli 1948*
Result	: *White by 1 point*

This game was played during one of the famous *chu-ban go*'s sponsored by the *Mainichi* newspaper.

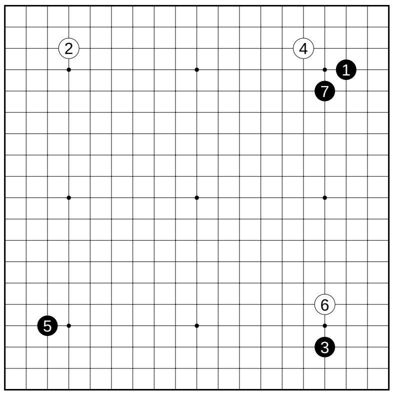


Figure 1: 1–7

White 4: A most common *kakari*.
Black 7: Typical *fuseki* in those days: the *Shusaku fuseki*.

Contents