

# xpat2 Version 1.05 Manual

Heiko Eiβfeldt      Michael Bischoff

12-Mar-1996

## Contents

# 1 Introduction

## 1.1 `xpat2` — A Generic Patience Game for X11

`xpat2` was created by merging the `Spider` patience game<sup>1</sup>, the `xsol` solitaire game (also known as `Klondike` under DOS) and the rules of some other patience games. Its main features are variable rule sets and different card sets for different resolution monitors. `xpat2` is the successor of `xpat`. New features include the Athena Widget interface and the possibility to configure keyboard commands and messages to national languages. `xpat2` looks best when you replace the standard `Xaw` library by its `Xaw3d` replacement.

This manual covers the Athena widget interface version of `xpat2`. It is possible to compile `xpat2` with the Xlib interface from `xpat`, but then some command line options, resource names, and other details may be different. Online help and button text customisation is not available with the Xlib interface.

The default cards of `xpat2` require the `Xpm` library and a display capable of displaying at least 128 colours (or shades of grey). For smaller resolutions (less than 800 \* 600) or monochrome X-servers you can compile `xpat2` with the old card graphics module, which offers a selection of smaller cards. For all options concerning the old module, please consult the `xpat` manual.

## 1.2 Quickstart, or How to Avoid Reading this Manual

If you already know `Spider` and don't want to read this documentation just type

```
xpat2 -rules Spider
```

to start `xpat2` with `Spider` rules. Type

```
xpat2 -?
```

to see the possible command-line options. If you are new to this game, and want to know more, read on.

---

<sup>1</sup>Copyright (c) 1989, Dave Lemke, Heather Rose, Donald R. Woods and Sun Microsystems, Inc.

## 1.3 Naming Conventions

Card *suits* are Clubs, Spades, Hearts, and Diamonds. The *ranks* go from Ace to King. There are different types of places where cards lie, or where you may place cards. We call all such locations *piles*. One pile contains the cards which have not been distributed yet. This is the *deck* or *hand*. The piles where some cards are put at the beginning of a new game, and where you move them, are called *slots*. Finally, the piles where you place the sorted arrays of all ranks and of the same suit are the *stacks* or *foundations*.

## 2 Special Rulesets

As mentioned before, `xpat2` knows about different patience rules. You can choose one of them by typing

```
xpat2 -rules (ruleset)
```

at the command prompt, or by simply linking `xpat2` to `spider`, `gypsy`, `freecell`, or `klondike` and calling the program with one of those names.

### 2.1 Rules for Spider

Remark:

- This section is adapted from the `Spider` documentation. See the file `COPYRIGHT.Spider` for details.

The object of the game is to build a sequence of cards of the same suit from King to Ace and move it from the slots to the stacks above. Once all eight sequences have been built and removed, you won the game. A more challenging version of this is to leave all eight sequences on the table until done, instead of removing those that are complete to the stacks above.

One may move the next lower card onto a card of the same or different suit; however, one may only move contiguous cards of the same suit as a group. For example, one may move the 5 of spades into either a 6 of spades or a 6 of hearts. Moving the 5 of spades onto the 6 of spades is a better move since now the 5–6 of spades group may be moved as a unit whereas the 5 of spades, 6 of hearts group may not. Also, moving the 4 of spades onto the 6

of spades is not a legal move. A King may only be moved into an empty slot (or removed to the stacks above when the sequence King–Ace is complete) since there are no higher cards than a King.

For general strategy, one wants to try to create a *space* since this is the most flexible way to move cards around. A space is where all cards in one column of the table have been removed. Note: before dealing the next round, all spaces have to have at least one card in them.

Also, when moving the cards, it is generally to your advantage to move the highest cards first and the lowest cards last doing the obvious moves first. For example, move a 5 of spades onto a 6 of spades, then Queen of hearts onto a King clubs before you move the 7 of clubs onto the 8 of hearts. Then you may choose to move 5–6 of spades group onto the 7 of clubs. The strategy is to combine the cards into same suit groups that may be moved as a unit. Once the group 5–6 of spades has been created, it may be broken up by using the middle mouse button on the lower card.

### **The Initial Tableau**

Shuffle together two decks of cards (104 cards in all) and deal ten cards face down in a row. Deal three more rows face down on the first. Next deal one card face down on each of four piles (traditionally the leftmost four, but it doesn't matter; for aesthetic reasons some people prefer the first, fourth, seventh, and tenth pile), for a total of 44 face-down cards. Finally, deal one card face up on each pile. These 54 cards constitute the initial tableau. NOTE: In the description below, the card at the *bottom* of a pile is taken to be the last one dealt or played onto that pile, as opposed to the card that is underneath all the others in the pile. Thus the *bottom card* is the one displayed bottommost on the screen by `xpat2`.

### **Building**

All building is done in the tableau, there being no separate foundations. On the bottom card of a pile may be placed any card of next-lower rank, regardless of suit. (Cards rank King (highest), Queen, Jack, 10, 9, . . . , 3, 2, Ace.) The bottom card of a pile can always be moved, as can any sequence of cards at the bottom of a pile that are consecutive and ascending in rank and of the same suit.

For example, suppose the 6, 5, and 4 of hearts are together at the bottom of a pile, with the 4 bottommost. They may be moved as a unit, or the 4 can be moved by itself, or the 5 and 4 can be moved without moving the 6. If the 6, 5, and 4 were moved onto a 7 of hearts at the bottom of some other pile, the four cards could then be moved as a unit onto any 8; if the 7 were not a heart, however, then once the hearts were placed upon it it would not be available to be moved until the hearts had been moved from it (to another 7 or into a space; see below).

When all face-up cards have been removed from a pile, the bottom face-down card is turned up and becomes available for play.

## Spaces

When all cards have been moved away from some pile, the resulting empty pile is called a *space* or *hole*. Any card or sequence of cards available for moving may be moved into a space. A King, or a sequence headed by a King, can *only* be moved into a space, and once moved there can never be moved out (except into another space, which doesn't accomplish anything) unless it is being removed entirely as described below.

## The Deck

The 50 cards not dealt initially form the *deck*. Whenever you wish (typically, whenever you get stuck), you may deal a new row of ten cards from the hand, face-up, upon the piles. NOTE: You are not allowed to do this if you have any spaces. You must fill them in first. Notice that these additional deals tend to introduce discontinuities in the piles; that is, you can get cards covering others that are not next-higher in rank. If you get stuck after having dealt the last of the five additional deals, you lost.

## Object of Play

When you have assembled a complete suit of thirteen cards, in sequence from King down to Ace, at the bottom of a pile, you may remove the thirteen cards from the tableau entirely. Cards so removed are never brought back into play; thus it is not always desirable to remove a suit when you have the opportunity, since it may pay to keep it around to aid in manipulating the

other cards of that suit (recall that there are 26 cards in each suit). The game is won if you manage to remove all eight suits.

If you find that a game is going so well that you're sure you're going to win, you can spice things up by trying to finish with as many completed suits as possible still in the tableau. Hardest of all is to finish the game with each of the eight suits brought together in sequence from King down to Ace, with all of the cards still in the tableau. Note that, once you start striving toward such a goal, you may make such a mess of the position that you won't be able to win at all!

## Scoring

Spider is a difficult game to master, and some players like to be able to evaluate their progress by scoring unsuccessful games. There are no rules for scoring in the literature, but the program implements the following rather arbitrary formula: 10 points for each initially face down card that gets turned over; 15 additional points for each column where all the face down cards have been turned over (even if you don't manage to get a space); 2 points for each card that is sitting atop the next higher card of the same suit; 50 points for each completed suit (in which case you do not also score for the 12 cards sitting atop next higher cards). This yields a maximum score of 990. If you win the game with 4 or more completed suits still in the tableau, add 2 points for each suit after the first three. Thus winning with all eight suits still in the tableau yields a score of 1000.

With the option “`-relaxed 1`”, you may deal new cards if all nonempty slots are in correct order, i.e. with respect to rank *and* suit. With the option “`-relaxed 2`”, you may deal new cards if all nonempty slots are in relaxed order, i.e. only with respect to rank. The purpose of this option is, that you do not need to destroy existing sequences before getting new cards.

## 2.2 Rules for Gypsy

(These rules are described by giving the differences to the **Spider** rules.) **Gypsy** is played with two decks of 52 cards. You have eight slots, and at the start of the game, three cards are dealt to each slot. Two cards lie facedown, and the bottom card is faceup.

One may move the next lower card onto a card only if their colours differ. For example, one may move the 4 of Clubs onto the 5 of Hearts or the 5 of Diamonds. The same rule applies for moving consecutive groups of cards: a group of cards can be moved as a block if the ranks of the cards are decreasing and their colours alternate.

Cards are moved to the stacks one at a time, in increasing rank, starting with the Ace. A card can only be moved to a stack of its suit. It is a good choice to move all Aces to the stacks immediately. It can be wiser, however, to keep cards of higher ranks on the slots, so you can put other cards on top of them. Cards can be moved back from the stacks to the slots (*borrowing*) if they follow the above rule to append. This can be useful if you want to create empty slots in order to move Kings there. (You may move *any* card to an empty slot, but empty slots are the only places where Kings may be moved to.)

## Scoring

**Gypsy** scores from 0 to 1000. There are two points of score for each alternating card pair on the slots and there are five points for each card on the stacks. Every stack which is filled from Ace to King adds another 60 points to the score.

Remark: The rules for **Gypsy** are no invention from the authors of `xpat2`, but the name **Gypsy** is. We have been playing **Gypsy** for years now and just don't know what the rules were originally called. Should you happen to know, tell us.

## 2.3 Rules for Klondike

**Klondike** is the traditional patience game and is played using only one set of cards. We have four stacks and seven slots. Initially, the first slot has one card on it, the second one two, and thus continuing up to the seventh slot, which gets seven cards. Of these cards, only the topmost card is turned face-up<sup>2</sup>. **Klondike** introduces another type of pile: There is a second deck which holds face-up cards. Every time you request the dealing of new cards, these cards are moved onto the open deck pile. You can move cards from

---

<sup>2</sup>You may specify the number of cards that are turned each time using the `-turn` option.

this pile to the slots or to the stacks, but you never can move cards back to the open deck. The rules which card may be grouped together are similar to the rules for the **Gypsy** game. The only difference is that you may only move a King onto an empty slot.

With the default rules, you may not move cards from the stacks back to the slots, but if you start `xpat2` with the `-relaxed 1` option, this is a valid move.

Once all cards are dealt, you can flip all the cards of the open deck back onto the other deck and continue. You can flip the deck up to three times. (This limit can be increased using the `-flips` option)

## 2.4 Rules for FreeCell

This variant has been shamelessly borrowed from the Microsoft Windows entertainment package. The rules introduce a new pile type, which we call *register*. A register may hold only one card at a time, but this may be an arbitrary one. This is to compensate for the fact that all cards are dealt at the start of game, which makes it very difficult to achieve a free slot. The cards dealt initially do not often form valid groups. You have to rearrange them, possibly using the registers as scratch storage. To get a free slot you have to build groups of cards which alternate in colour and have decreasing ranks.

The rules are easy to understand if you know **Gypsy** already since the rules for moving cards are the same. **FreeCell** is played with one deck of cards. You have four registers, four stacks, and eight slots.

The original **FreeCell** game has yet another restriction. When moving blocks of  $n$  cards, you must have at least  $n - 1$  registers free (moves are supposed to be single-card moves). This restriction can be dropped by calling `xpat2` with the `-relaxed 1` option, since it implies that only blocks of five cards maximum can be moved.

## 2.5 Rules for Seahaven Towers

**Seahaven Towers** is similar to **FreeCell** (or the other way round, I don't know which game was the first to exist). We have ten slots and four registers, but it is more difficult since cards may only be moved onto cards of the same suit and you can only move Kings to empty slots. As with **FreeCell**, the



original **Seahaven Towers** game has the restriction that, when moving blocks of  $n$  cards, you must have at least  $n - 1$  registers free (moves are supposed to be single-card moves). This restriction can be dropped by calling `xpat2` with the `-relaxed 1` option, since it implies that only blocks of five cards maximum can be moved.

## 2.6 Rules for Idiot's Delight

Idiot's Delight uses a standard 52-card deck (no jokers) with aces ranked high, above Kings. The game starts by dealing four cards in a row. You may discard any card that is lower than another card of the same suit. When no more cards can be removed, deal four more cards onto the remainder of the previous deal and remove all possible cards. If a pile is completely depleted, you can move a card from the top of another pile into the space.

The game continues in this way, dealing four cards at a time, then removing all possible cards. The game is won if only the four aces are left at the end of the game.

Be very careful of which card you move into spaces. The natural inclination is to move aces into spaces as soon as possible, but often it is advisable to consider other possibilities, depending on what cards can be freed up with each move.

## 2.7 Rules for Monte Carlo

Monte Carlo uses a standard 52-card deck (no jokers). To begin the game the deck is shuffled, then dealt face-up, one card at a time, into five rows of five cards each (the tableau), with the remainder of the deck held in reserve for later play.

The object of the game is to remove pairs of cards from the tableau. These cards can be any combination of colours, but must be of the same rank (eg: a 3 of hearts and a 3 of spades), and must be located adjacent to each other in a row, column or diagonal. Note that the removal of a card does not make the two cards beside it "adjacent".

After removing all possible pairs, the tableau must be consolidated, which means that the remaining cards in the tableau must be "backed up" (moved to the left, then up, keeping the original order) to fill in the blank spaces created by the removed pairs. Cards from the reserve are then dealt into the

newly created spaces and the process of removing pairs begins again. The tableau can be consolidated and re-dealt as many times as necessary.

The game is won when the entire deck has been removed, or lost when no more moves can be made.

## 2.8 Rules for Midnight Oil

Midnight Oil uses a standard 52-card deck (no jokers). To begin the game the deck is shuffled, then dealt face-up, one card at a time, into seventeen piles of three cards each. The last card is placed by itself as an eighteenth pile. These piles are also known as “fans”.

The object of the game is to move all of the cards from the eighteen fans into four new piles, arranged by suit, in order from Ace (low) to King (high). These four destination piles are known as the “foundations”. Only the top card of each fan is available for play.

In addition to moving cards from fans to foundations, cards can also be moved from one fan to another, building down in suit. In this way, submerged cards can be freed up for play to the foundations or to other fans. Once a fan becomes empty, no moves can be made to that fan.

When you have made all the moves you can, from the fans to the foundations and from fan to fan, the remaining cards in all the fans may be reshuffled and re-dealt in the same manner as at the beginning of the game (three cards to a fan). In all, the deck may be shuffled three times (once to begin the game, and twice more during the game).

At some point during the game, you may draw one submerged card from a fan and play it on any valid foundation or fan. It is generally better to save this draw for as long as possible.

## 2.9 Rules for Calculation

Calculation uses a standard 52-card deck (no jokers). To begin the game the deck is shuffled, then any ace, two, three, and four are pulled from the deck and form the start of four foundations. These foundations are built up according to the following sequence, regardless of suit or colour:

A	2	3	4	5	6	7	8	9	10	J	Q	K
2	4	6	8	10	Q	A	3	5	7	9	J	K
3	6	9	Q	2	5	8	J	A	4	7	10	K
4	8	Q	3	7	J	2	6	10	A	5	9	K

Cards are turned up from the deck one at a time, and can be played onto the foundations or onto any of four wastepiles located just below the foundations. The top of the wastepiles are always available for play to a foundation, but once a card is played to a wastepile, it cannot be moved again, except to a foundation.

The game is won by building all four foundations up to King. The skill involved in playing Calculation, in addition to just keeping track of what card goes next on each foundation, is in avoiding covering up a card you'll soon need with a card that cannot be used for some time, if at all. Kings are especially troublesome, since they cannot be moved until at least one foundation has been fully built up.

The general strategy is to build on the wastepiles in descending order, but the cards are seldom so kind as to allow this to any great degree.

## 2.10 Rules for Canfield

In Canfield, tableau cards may be built onto each other downward in alternate colours. An entire pile must be moved as a unit in building. Top cards of the piles are available for playing onto the foundations, but never into empty spaces.

Spaces must be filled from the stock. The top card of the stock also is available to be played on foundations or built on tableau piles. After the stock is exhausted, tableau spaces may be filled from the talon and the player may keep them open until he wishes to use them.

Cards are dealt from the hand to the talon by threes and this repeats until there are no more cards in the hand or the player quits. To have cards dealt onto the talon the player types “d” for this move. Foundation base cards are also automatically moved to the foundation when they become available.

The rules for betting are somewhat less strict than those used in the official version of the game. The initial deal costs \$13. You may quit at this point or inspect the game. Inspection costs \$13 and allows you to make as many moves as possible without moving any cards from your hand to the talon. (The initial deal places three cards on the talon; if all these cards are

used, three more are made available.) Finally, if the game seems interesting, you must pay the final installment of \$26. At this point you are credited at the rate of \$5 for each card on the foundation; as the game progresses you are credited with \$5 for each card that is moved to the foundation. Each run through the hand after the first costs \$5.

In `xpat2`, you have to pay the full amount of \$52 at the beginning. This makes a maximum score of \$208, if you use only one turn (i.e. no flips), which should be quite rare.

The layout of the window is as follows. In the first column, there are the stock, the talon, and the hand (from the upper window edge to the lower one). To the right of this column, the first row shows the four stacks (or foundations) and the second one the slots (tableau piles), which may grow downward.

Initially, you have 13 cards on the stock, 3 on the talon, and 31 on the hand. Every tableau pile consists of one card, and the last card, which defines the base rank, sits in one of the stacks. If you forget the base rank during the game, do a “restart game”, followed by an “undo”.

## 2.11 Rules for Royal Cotillion

Royal Cotillion uses two standard 52-card decks (no jokers) dealt together. The game begins by dealing twelve cards in three rows of four cards each, forming the left tableau. The right wing of the tableau is formed by dealing four rows of four cards each. The foundations are formed in two columns between the left and right tableau.

Foundations are formed in the following sequences for each suit:

A	3	5	7	9	J	K	2	4	6	8	10	Q
2	4	6	8	10	Q	A	3	5	7	9	J	K

In the left wing of the tableau, only the bottom card of each column is available for play, and spaces are never filled. In the right wing, all cards are available, and spaces are filled immediately from the wastepile, or, if there is none, from the deck.

Cards are turned up from the deck one at a time, and may be played either to the foundations or the wastepile. The top card of the wastepile is always available for play.

The game is won if all eight foundations are completed, or lost if a block is reached.

It is advisable, if you have a choice, to play from the left wing first, thus freeing up additional cards. In the right wing, you should not make a move merely because you can. It is better to wait to make a space until a desirable card to save turns up.

## 2.12 Rules for Baker's Dozen

You have 13 piles of 4 cards each, in which Kings are, if necessary, moved to the last position. The goal is to put all cards on the foundation, in order Ace to King. You are allowed to move all cards from the top of the piles (and from the top of the foundations) onto a pile where the last card is one more than the moved one, or onto the foundations (so a 5 of hearts could go under any six, or above the 4 of hearts in the foundation). Holes cannot be filled (this is why Kings are put on bottom of the piles, as they can be moved only onto the foundations). I like it, because it's rather quick, has a decent probability of success, and it's not automatic.

Baker's Dozen may be customized by changing the number of slots (in the range 2...13). With fewer slots, the cards of lower rank are already moved to the stack, which makes the game easier.

## 2.13 Rules for Oonsoo

Although the idea of Oonsoo is quite similar to the idea of Spider, it may be a bit confusing in the beginning, since in Oonsoo, you have to stack cards of the same rank. Oonsoo is played with 48 cards, giving 12 piles of 4 cards each to build. You may only move cards of suit club to empty spaces. The suits in descending order are Clubs, Spades, Hearts, Diamonds. You may only move cards of lower suit onto higher ones, thus the six of hearts may be placed upon the six of clubs and on the six of spades, but on no other card.

If you are interested in Oonsoo, please try also the Oonsoo implementation of Bradford W. Mott ([bwmott@eos.ncsu.edu](mailto:bwmott@eos.ncsu.edu)), which features special card graphics.

## 2.14 Customized Rules

You can customize the standard rules by giving the command line options

<code>-slots</code>	<i>slots</i>	to change the number of slots
<code>-decks</code>	<i>decks</i>	to change the number of decks
<code>-faceup</code>	<i>num</i> or	
<code>-facedown</code>	<i>num</i>	to set the initial number of face up / face down cards per slot (Spider and Gypsy only)
<code>-jokers</code>	<i>jokers</i>	to have some jokers
<code>-relaxed</code>	<i>num</i>	to set game-specific parameter 0
<code>-flips</code>	<i>num</i>	to set game-specific parameter 1
<code>-turn</code>	<i>num</i>	to set game-specific parameter 2
<code>-rotations</code>	<i>num</i>	to set game-specific parameter 3

The numerical arguments to the above options must lie in a certain range (typically 0 to 9 or 0 to 99), otherwise the option will be ignored silently. The options are ignored for any rule set which doesn't support them, without notifying the user.

- Customizing the rules will change the maximum score. Currently, it actually does disable scoring.
- Jokers are an experimental feature. An image is supplied with the internal cardset only. Use them if you want to, but there are no well-developed rules for patience with jokers.

Note that hints will not display every possible move with jokers (that would give too many). The hint algorithm assumes that you know that you can put a joker on *any* slot. With jokers, you can move *very* long groups, since they paste together subsequences.

Jokers may not go onto the stacks. A game is finished when all ordinary cards are moved to the stacks.

## 3 Keyboard and Mouse Commands

### 3.1 Mouse Commands

Three buttons of the mouse are supported. The actions depend on where you press a button.

When the mouse pointer is in the area of a card pile, the left mouse button (Button 1) does a quick move (if possible) and is only valid for slots. That is, a set of cards is moved to an automatically selected destination pile. This can either be a stack or another slot. If the destination is a stack, the vertical position of the pointer is ignored. If the destination is a slot, a maximal number of cards is moved, but the position of the pointer limits the selection. If, for example, the cards ten, nine and eight lie on top of a slot, and the mouse points to the nine, the eight and possibly the nine are selected to move (given that the colours match), but not the ten. Movement to stacks has priority over movement to other slots.

If you want to select the destination slot yourself, use the middle button. This will select a set of cards. The selected cards should appear shaded. (Or, if you like, are surrounded by a dashed or coloured line. See options `-markcolor`, `-markwidth`) You may now move to the destination and press the middle button again. Now the move will be done, if legal. To abort the procedure, just press on the source pile again. This command is also useful to see the length of sequences on the slots. To see the largest sequence to move, press the middle button when the mouse pointer is over the first dealt card in the column. It will then be marked (to unmark press it again).

After you have played a while, you will notice that the slots grow. When the bottom border of the window is encountered, the cards are shifted more closely together, such that they fit on the window. At some critical point, you cannot see what cards are on the slot any more (except for the card on top of the slot), because the cards are shifted too closely together. Move the pointer to the card you want to identify and press the right button. The whole card will be exposed to the foreground. It will disappear when you release the button. Hold down the button and move up and down the slot. The exposed cards are updated immediately as you move.

To use special commands (undo, redo, new game, quit) press the left button when the pointer is in the area of the appropriate text field on the top. Commands which can't be undone require a confirmation in the middle

of the window.

## 3.2 Keyboard Commands

Some commands can also be used with the keyboard, but you cannot play the game with a keyboard alone.

There are three classes of keyboard bindings: Built-in keyboard assignments have the least priority. Only the following commands are built-in:

key	action
Ctrl-L	redraw the window
<ESC>	abort a move
“B”	switch to Baker’s Dozen rules
“C”	switch to Canfield rules
“F”	switch to FreeCell rules
“G”	switch to Gypsy rules
“H”	switch to Seahaven rules
“I”	switch to Idiot’s Delight rules
“K”	switch to Klondike rules
“M”	switch to Monte Carlo rules
“O”	switch to Midnight Oil rules
“R”	switch to Royal Cotillion rules
“S”	switch to Spider rules

These commands can be extended or overridden by the global keyboard table, which may be different for every national language. The default assignment is



key	action
“y”	confirm action
“n”	abort action
“Q”	quit game (confirmation requested)
“d”	deal cards
“D”	deal cards
“A”	restart game
“N”	another game
“i”	information
“m”	do all obvious moves
“s”	show current score
“v”	print the version number of <code>xpat2</code>
“+”	suggest a move (forward)
“-”	suggest a move (backward)
“ ”	do the suggested move
“M”	resize the window
“P”	resize the window

With this table, the built-in binding of “M” is overridden.

At last, you may define local key bindings for different rules. These bindings have the highest priority. This feature is currently used only to rebind the rule-switching built-in commands to the “another game” function, which has the same effect, but avoids the recalculation of the screen layout.

## 4 Layout

`xpat2` supports dynamic layout of the window for some patience rules, for example `Gypsy` and `Spider`. Basically, there are two types. The first one we call the *big* layout. This can be characterized by the fact that the stacks are placed to the right of the slots, whereas in the *small* layout, they are above the slots<sup>3</sup>. The advantage of placing them at the side is, that the slots can grow larger in vertical direction, so you do not need to expose cards so often. The disadvantage is, that the window needs more space, except when you have many stacks. Initially, a minimum window size is determined by the program. This window size is large enough for one of the layout types. The

<sup>3</sup>Actually, there are 3 smaller layouts which differ slightly.

initialization routine computes the area needed for either layout type and chooses the one with smaller space. If you resize the window, the layout is changed dynamically.

At the command line, you can choose to have smaller or greater gaps between the cards. (`-gapx` and `-gapy` option, initially both are set to 4.) With the default values, `Spider` can be played in the large layout on a  $1024 \times 768$  display.

## 5 Cards

`xpat2` can be compiled with two different backends to display cards. One backend is identical to the version used in `xpat` and is contained in the file `X-gfx1.c`, the other one `X-gfx2.c` is new.

### 5.1 The Old Backend

#### Built-in Cards

`xpat2` supports different card sets without the need to recompile. One set of cards is built-in, so you can play `xpat2` without the need of carrying additional files. The built-in cards have size  $79 \times 123$  and are basically the large cards from the `Spider`-game, with some minor modifications to make them more symmetric.

#### 5.1.1 External Card Sets

If you have low resolution graphics, you can choose other card sets. Type

```
xpat -cards /usr/games/lib/xpat/small.cards
```

or simply

```
xpat -cards small
```

to use the external card set `/usr/games/lib/xpat/small.cards`. These cards are the small cards from `Spider`, they have a resolution of  $40 \times 60$ .

There is a third card set, `french.cards`, of resolution  $60 \times 93$  for medium resolution displays. The images of this card set are inspired by the “Französisches

Blatt”, the type of cards which are used in Germany<sup>4</sup>. The letters “B”, “D”, and “K” stand for “Bube”, “Dame”, “König”.

## Colour Cards

On colour displays, you may want to have more than only red and black for the card images. By using the Xpm-library of Arnaud Le Hors (Groupe Bull) and its  `pixmap`  editor, a colour version of  `bitmap` , the images of the King, Queen and Jack cards and the cardback were coloured. Every card of rank Jack, Queen, or King has an  `xpm` -file. The directory for the  `xpm` -files is given by the  `-xpmdir`  option.

The card images are no longer supplied (they were with  `xpat`  version 1), since the new backend replaces (and enhances) this function. Be patient, loading the colour cards will take some time.

Note: The Xpm-package version 3.3 or better supports GNU  `gzip`  compressed images, so you can  `gzip`  all  `.xpm` -files. This nice feature saves a lot of space on your hard disk.

## 5.2 The New Backend

This module requires the Xpm-library and works best on a server with at least 128 colours. With the  `-xpmdir`  option, several card sets may be selected. The file  `Cards.conf`  in the selected directory describes the details of the cards. The first line contains the overall size of the cards in pixels, the second line the number of different cardbacks and the number of jokers. If there is more than one cardback,  `xpat2`  randomly selects one of them during initialisation. The cardback is selected from one of the files  `Cardbackn.xpm` . All further lines of the file  `Cards.conf`  describe details of the card appearance, they are documented in the source.

If there is an image  `Background.xpm` , it is mapped onto the cards before the rank and suit symbols are drawn. By default, an X logo is shown. Replace it with your own silhouette, if you like.

---

<sup>4</sup>These are not the cards which are called “Deutsche Spielkarten”. They have even different suit symbols and are used seldom today.

## 6 Bells, Whistles, and Doodads

### 6.1 Internationalisation

The Athena widget set allows the redefinition of button labels by resource files. All other texts have also been made reconfigurable. The X-window system uses the contents of the LANG environment variable to select the path for the resource file (application defaults file). Just for fun, try

```
export LANG=german; xpat2
```

or

```
export LANG=italian; xpat2
```

or

```
export LANG=french; xpat2.
```

Please note that the keyboard bindings may also change (confirm may be “j”, or “s”, or “o”, then). Other languages are in preparation.

### 6.2 Saving Games

You can save the current state of the game by pressing the left mouse button on the “save game” button. `xpat2` will try to create a file with its name consisting of the name of the rule set as base name and the current seed as extension. If this name is too long for your filesystem, the base name is truncated. If the file did exist, it will be overwritten. Error messages will appear on `stderr`.

### 6.3 Logging of Successful Played Games

`xpat2` tries to log successfully solved games in `/var/games/xpat.log`. Create that file with mode 0666, if you want that. (Or, more UNIXy: create user `xpat`, `chown xpat2 xpat`, make `xpat` SUID, give write permissions to `xpat` only. Do as you like it.) To change the place for the log file, you have to recompile `xpat2`. Compile `loadsave.c` with

```
-DSCOREFILE=\"/home/mbi/xpat.log\"
```

to have the logfile in the directory `/home/mbi`.

## 6.4 Sound (technical)

### 6.4.1 FM Sound

On Linux systems, you may include support for FM sound or for digitized sound into `xpat2`. You need Hannu Savolainen's sound-kit version 2 or better<sup>5</sup>.

To use FM sound, you have to load an instrument into midi-channel 9 of the kernel (some type of bell, usually). Then you will hear these bells ringing when you successfully played a game. FM sound support uses the device `/dev/sequencer`.

### 6.4.2 Digitized Sound

For Linux and Suns, you can play u-law-encoded digitized sound files. Make sure that `/dev/audio` is writable for everyone. The directory for the sound files (its default is `/usr/games/lib/xpat`) is compiled into the module named `X-sound.SUN.o`. The directory is given by the `LIBDIR` macro, which must be defined when you compile `X-sound.SUN.c`.

Technically, the sound module just copies an audio file to `AUDIO_DEVICE`, which is `/dev/audio` by default, but may be changed at compile-time. If your sound device has another name, you give an alternate definition in the `Imakefile`. If you want to play WAV-files, or other formats which require an explicit program to play them, you may direct `AUDIO_DEVICE` to a named pipe, for example `/tmp/audio`, and start a background program reading this pipe. A sample WAV-player, `etc/wavplay.c`, is included in the source distribution, but as sound isn't basic to `xpat2`, you're on your own if you want to use this or a similar program.

## 6.5 Hints

`xpat2` offers the possibility to suggest a move. When you hit the `Hint` button the first time, possible moves are collected and valued using a very simple algorithm. Then the moves are sorted with respect to descending value. If you hit the `Hint` button with the left mouse button, the most valuable move is displayed. Upon the next press, you'll see the second hint. Once all hints

---

<sup>5</sup>With version 1 of the sound kit, you'll have to change `#include <sys/soundcard.h>` into `#include <linux/soundcard.h>` in the file `X-sound_FM.c`.

have been displayed, the text (**wrapping**) is displayed in the message window and the cycle is restarted. If you want to see a hint again, you can use the right button to cycle backwards through the ring buffer. If you want to do a move which is currently shown, simply press the middle mouse button in the **Hint** button.

**ATTENTION!** Moves to empty slots are only displayed, if Kings are the only cards which may go there. Other moves, which are valued too bad are not displayed either, for example, any move that takes back cards from the stacks to the slots. Thus the message “no hints” does **not** mean, that there is no possible move!

There is a command **AllMoves**, bound to key “**m**”, which iteratively selects the best valued hint and performs it (a sort of short-distancs auto-play). Be careful with this one, as it can trash your game!

## 6.6 Taking Back Moves

**xpat2** logs every move. This gives you the possibility of unlimited undo. Since it is not nice, however, to turn cards and put them back if you do not like what’s underneath them, taking back a move which gave you new information is considered a cheat, and logged. To restart a game is considered a big cheat, since you may now know the positions of ALL cards. Starting **xpat2** with a given random seed, which has the same effect, is not considered to be a cheat, because it can be possible that a friend gave you the seed (try this one. . .).

Those who want to play honest can undo an accidently done cheat (redo a move), which decreases the cheat count again.

Cheat detection is not implemented for all rule sets.

## 6.7 Selecting a Random Number Generator

Unfortunately, the random permutations generated by **xpat2**, up to version 1.03, were not distributed uniformly. This causes a problem, since uniform distributions should be used, but changing the shuffle routine would imply that all previously saved games made by **xpat2** would be invalid now. To avoid this, you may select the old random number generator by the option

`-rng 0,`

but by default, a new random distribution is used by `xpat2` version 1.04 and up.

## 6.8 .Xdefaults Names

The following table gives an overview about the mappings of command line options to their corresponding Xdefaults names.

option	resource name	option	resource name
<code>-rules</code>	<code>rules</code>	<code>-xpmdir</code>	<code>xpmDir</code>
<code>-cards</code>	<code>cards</code>	<code>-tmps</code>	<code>tmps</code>
<code>-decks</code>	<code>decks</code>	<code>-slots</code>	<code>slots</code>
<code>-faceup</code>	<code>faceup</code>	<code>-facedown</code>	<code>facedown</code>
<code>-jokers</code>	<code>jokers</code>	<code>-rng</code>	<code>rng</code>
<code>-relaxed</code>	<code>relaxed</code>	<code>-flips</code>	<code>flips</code>
<code>-turn</code>	<code>turn</code>	<code>-rotations</code>	<code>rotations</code>
<code>-cb</code>	<code>cardbackColor</code>	<code>-tb</code>	<code>background</code>
<code>-arrowcolor</code>	<code>arrowColor</code>	<code>-markwidth</code>	<code>markWidth</code>
<code>-markcolor</code>	<code>markColor</code>	<code>-nosound</code>	<code>sound</code>
<code>-sound</code>	<code>sound</code>		

The resources are part of the `Tableau` widget.

Example: You may insert the line

```
XPat*Tableau.xpmDir: /usr/games/lib/xpat/funnycards
```

into your `.Xdefaults` file to have other cards. If you are missing the background colour and larger spaces between the cards from `Spider`, just insert

```
XPat*Tableau.background: SeaGreen
```

```
XPat*Tableau.Gap: 20
```

there.

## 6.9 Assigning Your Own Keybindings

`xpat2` reads a keyboard definition file (`/usr/games/lib/xpat/keys` by default), which you may customize. The structure of the file is easy: Every line contains a character (key), a space, and the name of the function to be assigned to this key. The valid function names are:

name	description
None	no operation
rq_LeavePat	quit (confirmed)
rq_RestartGame	restart (confirmed)
rq_AnotherGame	another game (confirmed)
LeavePat	quit (unconditionally)
RestartGame	restart (unconditionally)
AnotherGame	another game (unconditionally)
ResetHints	reset the hint mechanism
DoHint	execute the hint currently displayed
NextHint	show the next hint
PreviousHint	show the previous hint
CancelSelection	abort a move
AllMoves	do all obvious moves
UndoMove	undo a move
RedoMove	redo a move
DealCards	deal cards, or flip the talon
ShowScore	show current score
ReplayGame	quick replay (gimmick only)
Layout	recompute the window layout
SaveGame	save current game
ShowVersion	show version number of <code>xpat2</code>
MinWindow	resize window to minimum size
PreferredWindow	resize window to preferred size
Info	more information on the game
ToStack	move all or selected cards to the stacks
OneToStack	move selected card to stack
AllToStack	move all cards to the stacks
DropBookmark	drop a bookmark
GotoBookmark	go to bookmark
RotateUp	rotate a slot up
RotateDown	rotate a slot down
Cancel	deny confirmation
Confirm	confirm action

Three commands are available in a secure version (which pops up a confirm window), and an unconditional version. The confirm window is omitted, if a



game is solved, or no move has been made yet. Actually, there is no need for the confirmed version of “RestartGame”, since this move can be taken back by the undo command.

The “MinWindow” and “Preferred Window” commands are an experimental feature. They may require to be called twice, due to changes in the command button area, which are not precomputed. The “MinWindow” function resizes the other window of `xpat2` to the size needed for the smallest layout. The “Preferred Window” function resizes the other window of `xpat2` to the size needed for the current layout.

## 6.10 Known Bugs

- There is a problem with X-servers which are short on memory. In this case, some parts of the `xpat2` window may be left blank after a resize event. You have to hit “ctrl-L” to get the window redrawn correctly.
- There is no test whether allocating a colour has succeeded. If you have `xv` running or some colourful background, `xpat2` may appear white-on-white. Some think, it’s a feature to make it more difficult :-)
- At startup, all the cards seem to be drawn twice.
- Loading of colour cards (`.xpm`-files) is too slow.

## 6.11 Unknown Bugs

Should you find an unknown bug, please e-mail a short description to

`mbi@mo.math.nat.tu-bs.de` or `heiko@colossus.escape.de`.

A bug is something which either gives weird effects or behaves differently as described in this manual.

## 6.12 Other Rules

Type

```
xpat2 -rules ?
```

to see the possible rules currently implemented in `xpat2`. Due to the well-known discrepancy of a program with its documentation, there may be more rules implemented than described here.

By the way, did you know that `Spider` is available under the name “Arachnid” for Microsoft Windows?

## 7 Copyright

Copyright © 1993, 1994 by Heiko Eißfeldt and Michael Bischoff  
([heiko@colossus.escape.de](mailto:heiko@colossus.escape.de), [mbi@mo.math.nat.tu-bs.de](mailto:mbi@mo.math.nat.tu-bs.de)).

Copyright © 1990 by David Lemke & Network Computing Devices, Inc.  
([lemke@ncd.com](mailto:lemke@ncd.com))

Copyright © 1990 Heather Rose and Sun Microsystems, Inc.

Copyright © 1989, Donald R. Woods and Sun Microsystems, Inc.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both the copyright notice and this permission notice appear in supporting documentation.

`xpat` and `xpat2` were developed under Linux, the free UNIX for the IBM-PC and compatibles. `xpat2` is based on `xpat`, which is based on the game `Spider`. All code which is not covered by `Spider` copyrights is distributed by terms of the GNU General public license (GNU CopyLeft). You should have received a copy of the GNU General public license in the file `COPYRIGHT.GNU`.