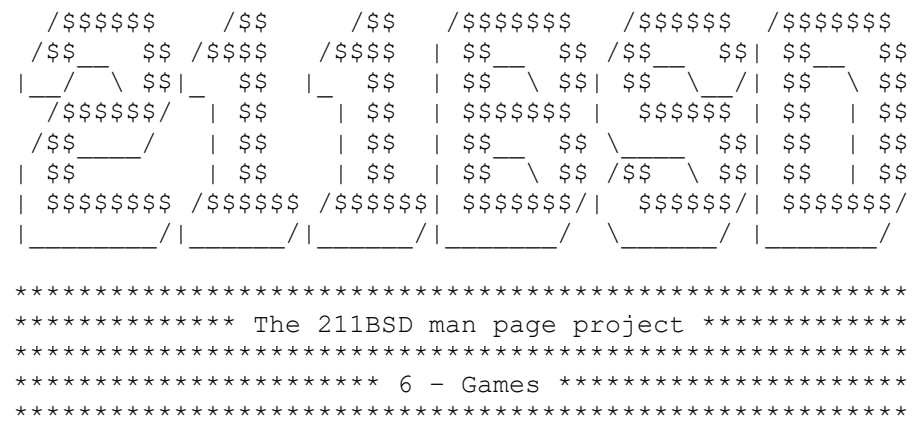



Inspired by:
==========

| | |
|---|---|
| SimH | http://simh.trailing-edge.com/ |
| PiDP11 | https://obsolescence.wixsite.com/obsolescence/pidp-11 |
| BSD 2.11 | https://wfjm.github.io/home/211bsd/ |

Presented by the ShadowTron Blog
==========

https://www.youtube.com/c/shadowtronblog
www.shadowtron.com
shadowtronblog <at> gmail <dot> com

Other manuals in the series
==========

    Manual 1  - Commands and Application Programs
    Manual 2  - System Calls
    Manual 3  - C Library Subroutines
    Manual 3F - Fortran Library
    Manual 4  - Special Files
    Manual 5  - File Formats
==> Manual 6  - Games
    Manual 7  - Miscellaneous
    Manual 8  - System Maintenance

**********
**** Manual 6 - Games ****
**********

| Page | Command | Description |
|---|---|---|
| 3 | aardvark | yet another exploration game |
| 4 | adventure | an exploration game |
| 5 | arithmetic | provide drill in number facts |
| 6 | backgammon | the game |
| 7 | banner | print large banner on printer |
| 8 | battlestar | a tropical adventure game |
| 11 | bcd | convert to antique media |
| 12 | bj | the game of black jack |
| 14 | boggle | play the game of boggle |
| 15 | canfield | the solitaire card game canfield |
| 17 | chess | the game of chess |
| 18 | ching | the book of changes and other cookies |
| 20 | cribbage | the card game cribbage |
| 22 | doctor | interact with a psychoanalyst |
| 23 | fish | play "Go Fish" |
| 24 | fortune | print a random, hopefully interesting adage |
| 25 | hangman | Computer version of the game hangman |
| 26 | hunt | a multi-player multi-terminal game |
| 30 | mille | play Mille Bournes |
| 35 | monop | Monopoly game |
| 38 | moo | guessing game |
| 39 | number | convert Arabic numerals to English |
| 40 | quiz | test your knowledge |
| 41 | rain | animated raindrops display |
| 42 | robots | fight off villainous robots |
| 44 | rogue | Exploring The Dungeons of Doom |
| 46 | sail | multi-user wooden ships and iron men |
| 63 | snake | display chase game |
| 65 | trek | trekkie game |
| 66 | ttt | tic-tac-toe |
| 67 | warp | a real-time space war game |
| 70 | words | word game |
| 71 | worm | Play the growing worm game |
| 72 | worms | animate worms on a display terminal |
| 73 | wump | the game of hunt-the-wumpus |
| 74 | zork | the game of dungeon |

This game is not included on the 211BSD image.

NAME

aardvark - yet another exploration game

SYNOPSIS

/usr/games/aardvark

DESCRIPTION

Aardvark is yet another computer fantasy simulation game of the adventure/zork genre. This one is written in DDL (Dungeon Definition Language) and is intended primarily as an example of how to write a dungeon in DDL.

FILES

/usr/games/lib/ddlrun ddl interpreter
/usr/games/lib/aardvarkinternal form of aardvark dungeon

AUTHOR

Mike Urban, UCLA

NAME

adventure - an exploration game

SYNOPSIS

/usr/games/adventure

DESCRIPTION

The object of the game is to locate and explore Colossal Cave, find the treasures hidden there, and bring them back to the building with you. The program is self-descriptive to a point, but part of the game is to discover its rules.

To terminate a game, type `quit'; to save a game for later resumption, type `suspend'.

BUGS

Saving a game creates a large executable file instead of just the information needed to resume the game.

NAME

arithmetic - provide drill in number facts

SYNOPSIS

/usr/games/arithmetic [ +-x/ ] [ range ]

DESCRIPTION

Arithmetic types out simple arithmetic problems, and waits for an answer to be typed in. If the answer is correct, it types back "Right!", and a new problem. If the answer is wrong, it replies "What?", and waits for another answer. After every twenty problems, it publishes statistics on correctness and the time required to answer.

To quit the program, type an interrupt (delete).

The first optional argument determines the kind of problem to be generated; +-x/ respectively cause addition, subtraction, multiplication, and division problems to be generated. One or more characters can be given; if more than one is given, the different types of problems will be mixed in random order; default is +-.

Range is a decimal number; all addends, subtrahends, differences, multiplicands, divisors, and quotients will be less than or equal to the value of range. Default range is 10.

At the start, all numbers less than or equal to range are equally likely to appear. If the respondent makes a mistake, the numbers in the problem which was missed become more likely to reappear.

As a matter of educational philosophy, the program will not give correct answers, since the learner should, in principle, be able to calculate them. Thus the program is intended to provide drill for someone just past the first learning stage, not to teach number facts de novo. For almost all users, the relevant statistic should be time per problem, not percent correct.

NAME

backgammon - the game

SYNOPSIS

/usr/games/backgammon

DESCRIPTION

This program does what you expect. It will ask whether you need instructions.

NAME

banner - print large banner on printer

SYNOPSIS

/usr/games/banner [ -wn ] message ...

DESCRIPTION

Banner prints a large, high quality banner on the standard output. If the message is omitted, it prompts for and reads one line of its standard input. If -w is given, the output is scrunched down from a width of 132 to n , suitable for a narrow terminal. If n is omitted, it defaults to 80.

The output should be printed on a hard-copy device, up to 132 columns wide, with no breaks between the pages. The volume is great enough that you may want a printer or a fast hardcopy terminal, but if you are patient, a decwriter or other 300 baud terminal will do.

BUGS

Several ASCII characters are not defined, notably <, >, [, ], \, ^, _, {, }, |, and ~. Also, the characters ", ', and & are funny looking (but in a useful way.)

The -w option is implemented by skipping some rows and columns. The smaller it gets, the grainier the output. Sometimes it runs letters together.

AUTHOR

Mark Horton

NAME

battlestar - a tropical adventure game

SYNOPSIS

battlestar [ -r (recover a saved game) ]

DESCRIPTION

Battlestar is an adventure game in the classic style. However, It's slightly less of a puzzle and more a game of exploration. There are a few magical words in the game, but on the whole, simple English should suffice to make one's desires understandable to the parser.

THE SETTING

In the days before the darkness came, when battlestars ruled the heavens...

> Three He made and gave them to His daughters,
> Beautiful nymphs, the goddesses of the waters.
> One to bring good luck and simple feats of wonder,
> Two to wash the lands and churn the waves asunder,
> Three to rule the world and purge the skies with thunder.

In those times great wizards were known and their powers were beyond belief. They could take any object from thin air, and, uttering the word

In those times men were known for their lust of gold and desire to wear fine weapons. Swords and coats of mail were fashioned that could withstand a laser blast.

But when the darkness fell, the rightful reigns were toppled. Swords and helms and heads of state went rolling across the grass. The entire fleet of battlestars was reduced to a single ship.

SAMPLE COMMANDS

take --- take an object
drop --- drop an object

wear --- wear an object you are holding
draw --- carry an object you are wearing

puton --- take an object and wear it
take off -- draw an object and drop it

throw <object> <direction>

! <shell esc>

IMPLIED OBJECTS

```
>-: take watermellon
watermellon:
Taken.
>-: eat
watermellon:
Eaten.
>-: take knife and sword and apple, drop all
knife:
Taken.
broadsword:
Taken.
apple:
Taken.
knife:
Dropped.
broadsword:
Dropped.
apple:
Dropped.
>-: get
knife:
Taken.
```

Notice that the "shadow" of the next word stays around if you want to take advantage of it. That is, saying "take knife" and then "drop" will drop the knife you just took.

SCORE & INVEN

The two commands "score" and "inven" will print out your current status in the game.

SAVING A GAME

The command "save" will save your game in a file called "Bstar." You can recover a saved game by using the "-r" option when you start up the game.

DIRECTIONS

The compass directions N, S, E, and W can be used if you have a compass. If you don't have a compass, you'll have to say R, L, A, or B, which stand for Right, Left, Ahead, and Back. Directions printed in room descriptions are always printed in R, L, A, & B relative directions.

HISTORY

I wrote Battlestar in 1979 in order to experiment with the niceties of the C Language. Most interesting things that happen in the game are hardwired into the code, so don't send me any hate mail about it! Instead, enjoy art for art's sake!

AUTHOR
David Riggle

INSPIRATION & ASSISTANCE
Chris Guthrie
Peter Da Silva
Kevin Brown
Edward Wang
Ken Arnold & Company

BUGS
Countless.

FAN MAIL
Send to edward%ucbarpa@Berkeley.arpa,
chris%ucbcory@berkeley.arpa, riggle.pa@xerox.arpa.

NAME

bcd, ppt - convert to antique media

SYNOPSIS

/usr/games/bcd text

/usr/games/ppt

DESCRIPTION

Bcd converts the literal text into a form familiar to old-timers.

Ppt converts the standard input into yet another form.

SEE ALSO

dd(1)

NAME

bj - the game of black jack

SYNOPSIS

/usr/games/bj

DESCRIPTION

Bj is a serious attempt at simulating the dealer in the game of black jack (or twenty-one) as might be found in Reno. The following rules apply:

The bet is $2 every hand.

A player `natural' (black jack) pays $3. A dealer natural loses $2. Both dealer and player naturals is a `push' (no money exchange).

If the dealer has an ace up, the player is allowed to make an `insurance' bet against the chance of a dealer natural. If this bet is not taken, play resumes as normal. If the bet is taken, it is a side bet where the player wins $2 if the dealer has a natural and loses $1 if the dealer does not.

If the player is dealt two cards of the same value, he is allowed to `double'. He is allowed to play two hands, each with one of these cards. (The bet is doubled also; $2 on each hand.)

If a dealt hand has a total of ten or eleven, the player may `double down'. He may double the bet ($2 to $4) and receive exactly one more card on that hand.

Under normal play, the player may `hit' (draw a card) as long as his total is not over twenty-one. If the player `busts' (goes over twenty-one), the dealer wins the bet.

When the player `stands' (decides not to hit), the dealer hits until he attains a total of seventeen or more. If the dealer busts, the player wins the bet.

If both player and dealer stand, the one with the largest total wins. A tie is a push.

The machine deals and keeps score. The following questions will be asked at appropriate times. Each question is answered by y followed by a new line for `yes', or just new line for `no'.

? (means, `do you want a hit?')
Insurance?

Double down?

Every time the deck is shuffled, the dealer so states and the `action' (total bet) and `standing' (total won or lost) is printed. To exit, hit the interrupt key (DEL) and the action and standing will be printed.

NAME

boggle - play the game of boggle

SYNOPSIS

/usr/games/boggle [ + ] [ ++ ]

DESCRIPTION

This program is intended for people wishing to sharpen their skills at Boggle (TM Parker Bros.). If you invoke the program with 4 arguments of 4 letters each, (e.g. " .}S 3 1 "" " "boggle" "appl" "epie" "moth" "erhd"" the program forms the obvious Boggle grid and lists all the words from /usr/dict/words found therein. If you invoke the program without arguments, it will generate a board for you, let you enter words for 3 minutes, and then tell you how well you did relative to /usr/dict/words.

The object of Boggle is to find, within 3 minutes, as many words as possible in a 4 by 4 grid of letters. Words may be formed from any sequence of 3 or more adjacent letters in the grid. The letters may join horizontally, vertically, or diagonally. However, no position in the grid may be used more than once within any one word. In competitive play amongst humans, each player is given credit for those of his words which no other player has found.

In interactive play, enter your words separated by spaces, tabs, or newlines. A bell will ring when there is 2:00, 1:00, 0:10, 0:02, 0:01, and 0:00 time left. You may complete any word started before the expiration of time. You can surrender before time is up by hitting 'break'. While entering words, your erase character is only effective within the current word and your line kill character is ignored.

Advanced players may wish to invoke the program with 1 or 2 +'s as the first argument. The first + removes the restriction that positions can only be used once in each word. The second + causes a position to be considered adjacent to itself as well as its (up to) 8 neighbors.

NAME

canfield, cfscores - the solitaire card game canfield

SYNOPSIS

/usr/games/canfield
/usr/games/cfscores

DESCRIPTION

If you have never played solitaire before, it is recommended that you consult a solitaire instruction book. In Canfield, tableau cards may be built on each other downward in alternate colors. An entire pile must be moved as a unit in building. Top cards of the piles are available to be played on 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 'ht' for his move. Foundation base cards are also automatically moved to the foundation when they become available.

The command 'c' causes canfield to maintain card counting statistics on the bottom of the screen. When properly used this can greatly increase one's chances of winning.

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. The card counting feature costs $1 for each unknown card that is identified. If the information is toggled on, you are only charged for cards that became visible since it was last turned on. Thus the maximum cost of information is $34. Playing time is charged at a rate of $1 per minute.

With no arguments, the program cfscores prints out the current status of your canfield account. If a user name is specified, it prints out the status of their canfield account. If the -a flag is specified, it prints out the canfield accounts for all users that have played the game since the database was set up.

FILES

/usr/games/canfield the game itself
/usr/games/cfscores the database printer
/usr/games/lib/cfscores the database of scores

BUGS

It is impossible to cheat.

AUTHORS

Originally written: Steve Levine
Further random hacking by: Steve Feldman, Kirk McKusick, Mikey Olson, and Eric Allman.

NAME

chess - the game of chess

SYNOPSIS

/usr/games/chess

DESCRIPTION

Chess is a computer program that plays class D chess. Moves may be given either in standard (descriptive) notation or in algebraic notation. The symbol `+' is used to specify check; `o-o' and `o-o-o' specify castling. To play black, type `first'; to print the board, type an empty line.

Each move is echoed in the appropriate notation followed by the program's reply.

FILES

/usr/games/lib/book opening `book'

DIAGNOSTICS

The most cryptic diagnostic is `eh?' which means that the input was syntactically incorrect.

WARNING

Over-use of this program will cause it to go away.

BUGS

Pawns may be promoted only to queens.

This game is not included on the 211BSD image.

NAME

ching - the book of changes and other cookies

SYNOPSIS

/usr/games/ching [ hexagram ]

DESCRIPTION

The I Ching or Book of Changes is an ancient Chinese oracle that has been in use for centuries as a source of wisdom and advice.

The text of the oracle (as it is sometimes known) consists of sixty-four hexagrams, each symbolized by a particular arrangement of six straight (---) and broken (- -) lines. These lines have values ranging from six through nine, with the even values indicating the broken lines.

Each hexagram consists of two major sections. The Judgement relates specifically to the matter at hand (E.g., "It furthers one to have somewhere to go.") while the Image describes the general attributes of the hexagram and how they apply to one's own life ("Thus the superior man makes himself strong and untiring.").

When any of the lines have the values six or nine, they are moving lines; for each there is an appended judgement which becomes significant. Furthermore, the moving lines are inherently unstable and change into their opposites; a second hexagram (and thus an additional judgement) is formed.

Normally, one consults the oracle by fixing the desired question firmly in mind and then casting a set of changes (lines) using yarrow-stalks or tossed coins. The resulting hexagram will be the answer to the question.

Using an algorithm suggested by S. C. Johnson, the UNIX oracle simply reads a question from the standard input (up to an EOF) and hashes the individual characters in combination with the time of day, process id and any other magic numbers which happen to be lying around the system. The resulting value is used as the seed of a random number generator which drives a simulated coin-toss divination. The answer is then piped through nroff for formatting and will appear on the standard output.

For those who wish to remain steadfast in the old traditions, the oracle will also accept the results of a personal divination using, for example, coins. To do this, cast the change and then type the resulting line values as an argument.

The impatient modern may prefer to settle for Chinese cookies; try fortune(6).

SEE ALSO

It furthers one to see the great man.

DIAGNOSTICS

The great prince issues commands,
Founds states, vests families with fiefs.
Inferior people should not be employed.

BUGS

Waiting in the mud
Brings about the arrival of the enemy.

If one is not extremely careful,
Somebody may come up from behind and strike him.
Misfortune.

NAME

cribbage - the card game cribbage

SYNOPSIS

/usr/games/cribbage [ -req ] name ...

DESCRIPTION

Cribbage plays the card game cribbage, with the program playing one hand and the user the other. The program will initially ask the user if the rules of the game are needed - if so, it will print out the appropriate section from According to Hoyle with more (I).

Cribbage options include:

-e When the player makes a mistake scoring his hand or crib, provide an explanation of the correct score. (This is especially useful for beginning players.)

-q Print a shorter form of all messages - this is only recommended for users who have played the game without specifying this option.

-r Instead of asking the player to cut the deck, the program will randomly cut the deck.

Cribbage first asks the player whether he wishes to play a short game ("once around", to 61) or a long game ("twice around", to 121). A response of `s' will result in a short game, any other response will play a long game.

At the start of the first game, the program asks the player to cut the deck to determine who gets the first crib. The user should respond with a number between 0 and 51, indicating how many cards down the deck is to be cut. The player who cuts the lower ranked card gets the first crib. If more than one game is played, the loser of the previous game gets the first crib in the current game.

For each hand, the program first prints the player's hand, whose crib it is, and then asks the player to discard two cards into the crib. The cards are prompted for one per line, and are typed as explained below.

After discarding, the program cuts the deck (if it is the player's crib) or asks the player to cut the deck (if it's its crib); in the latter case, the appropriate response is a number from 0 to 39 indicating how far down the remaining 40 cards are to be cut.

After cutting the deck, play starts with the non-dealer (the person who doesn't have the crib) leading the first card.

Play continues, as per cribbage, until all cards are exhausted. The program keeps track of the scoring of all points and the total of the cards on the table.

After play, the hands are scored. The program requests the player to score his hand (and the crib, if it is his) by printing out the appropriate cards (and the cut card enclosed in brackets). Play continues until one player reaches the game limit (61 or 121).

A carriage return when a numeric input is expected is equivalent to typing the lowest legal value; when cutting the deck this is equivalent to choosing the top card.

Cards are specified as rank followed by suit. The ranks may be specified as one of: `a', `2', `3', `4', `5', `6', `7', `8', `9', `t', `j', `q', and `k', or alternatively, one of: "ace", "two", "three", "four", "five", "six", "seven", "eight", "nine", "ten", "jack", "queen", and "king". Suits may be specified as: `s', `h', `d', and `c', or alternatively as: "spades", "hearts", "diamonds", and "clubs". A card may be specified as: <rank> " " <suit>, or: <rank> " of " <suit>. If the single letter rank and suit designations are used, the space separating the suit and rank may be left out. Also, if only one card of the desired rank is playable, typing the rank is sufficient. For example, if your hand was "2H, 4D, 5C, 6H, JC, KD" and it was desired to discard the king of diamonds, any of the following could be typed: "k", "king", "kd", "k d", "k of d", "king d", "king of d", "k diamonds", "k of diamonds", "king diamonds", or "king of diamonds".

FILES

/usr/games/cribbage

AUTHORS

Earl T. Cohen wrote the logic. Ken Arnold added the screen oriented interface.

This game is not included on the 211BSD image.

NAME

doctor - interact with a psychoanalyst

SYNOPSIS

/usr/games/doctor

DESCRIPTION

Doctor is a lisp-language version of the legendary ELIZA program of Joseph Weizenbaum. This script "simulates" a Rogerian psychoanalyst. Type in lower case, and when you get tired or bored, type your interrupt character (either control-C or Rubout). Remember to type two carriage returns when you want it to answer.

In order to run this you must have a Franz Lisp system in /usr/ucb/lisp.

AUTHORS

Adapted for Lisp by Jon L White, moved to Franz by John Foderaro, from an original script by Joseph Weizenbaum.

NAME

fish - play ``Go Fish''

SYNOPSIS

/usr/games/fish

DESCRIPTION

Fish plays the game of "Go Fish", a childrens' card game. The Object is to accumulate `books' of 4 cards with the same face value. The players alternate turns; each turn begins with one player selecting a card from his hand, and asking the other player for all cards of that face value. If the other player has one or more cards of that face value in his hand, he gives them to the first player, and the first player makes another request. Eventually, the first player asks for a card which is not in the second player's hand: he replies `GO FISH!' The first player then draws a card from the `pool' of undealt cards. If this is the card he had last requested, he draws again. When a book is made, either through drawing or requesting, the cards are laid down and no further action takes place with that face value.

To play the computer, simply make guesses by typing a, 2, 3, 4, 5, 6, 7, 8, 9, 10, j, q, or k when asked. Hitting return gives you information about the size of my hand and the pool, and tells you about my books. Saying `p' as a first guess puts you into `pro' level; The default is pretty dumb.

NAME

fortune - print a random, hopefully interesting, adage

SYNOPSIS

/usr/games/fortune [ - ] [ -wslao ]

DESCRIPTION

Fortune with no arguments prints out a random adage. The flags mean:

-w Waits before termination for an amount of time calculated from the number of characters in the message. This is useful if it is executed as part of the logout procedure to guarantee that the message can be read before the screen is cleared.

-s Short messages only.

-l Long messages only.

-o Choose from an alternate list of adages, often used for potentially offensive ones.

-a Choose from either list of adages.

FILES

/usr/games/lib/fortunes.dat

AUTHOR

Ken Arnold

NAME

hangman - Computer version of the game hangman

SYNOPSIS

/usr/games/hangman

DESCRIPTION

In hangman, the computer picks a word from the on-line word list and you must try to guess it. The computer keeps track of which letters have been guessed and how many wrong guesses you have made on the screen in a graphic fashion.

FILES

| | |
|---|---|
| /usr/dict/words | On-line word list |

AUTHOR

Ken Arnold

NAME

hunt - a multi-player multi-terminal game

SYNOPSIS

/usr/games/hunt [-q] [-m] [hostname] [-l name]

DESCRIPTION

The object of the game hunt is to kill off the other players. There are no rooms, no treasures, and no monsters. Instead, you wander around a maze, find grenades, trip mines, and shoot down walls and players. The more players you kill before you die, the better your score is. If the -m flag is given, you enter the game as a monitor (you can see the action but you cannot play).

Hunt normally looks for an active game on the local network; if none is found, it starts one up on the local host. One may specify the location of the game by giving the hostname argument. The player name may be specified on the command line by using the -l option. This command syntax was chosen for rlogin/rsh compatibility. If the -q flag is given, hunt queries the network and reports if an active game were found. This is useful for .login scripts.

Hunt only works on crt (vdt) terminals with at least 24 lines, 80 columns, and cursor addressing. The screen is divided in to 3 areas. On the right hand side is the status area. It shows you how much damage you've sustained, how many charges you have left, who's in the game, who's scanning (the asterisk in front of the name), who's cloaked (the plus sign in front of the name), and other players' scores. Most of the rest of the screen is taken up by your map of the maze, except for the 24th line, which is used for longer messages that don't fit in the status area.

Hunt uses the same keys to move as vi does, i.e., h,j,k, and l for left, down, up, right respectively. To change which direction you're facing in the maze, use the upper case version of the movement key (i.e., HJKL).

Other commands are:

f - Fire (in the direction you're facing) (Takes 1 charge)
g - Throw grenade (in the direction you're facing) (Takes 9 charges)
F - Throw satchel charge (Takes 25 charges)
G - Throw bomb (Takes 49 charges)
o - Throw small slime bomb (Takes 15 charges)
O - Throw big slime bomb (Takes 30 charges)
s - Scan (show where other players are) (Takes 1 charge)
c - Cloak (hide from scanners) (Takes 1 charge)

^L - Redraw screen

```
   q   - Quit
```

Knowing what the symbols on the screen often helps:

```
   -|+ - walls
   /\  - diagonal (deflecting) walls
   #   - doors (dispersion walls)
   ;   - small mine
   g   - large mine
   :   - shot
   o   - grenade
   O   - satchel charge
   @   - bomb
   s   - small slime bomb
   $   - big slime bomb
   ><^v      - you facing right, left, up, or down
   }{i!      - other players facing right, left, up, or down
   *   - explosion
   \|/
   -*- - grenade and large mine explosion
   /|\
```

Satchel and bomb explosions are larger than grenades (5x5, 7x7, and 3x3 respectively).

Other helpful hints:

[] You can only fire in the direction you are facing.
[] You can only fire three shots in a row, then the gun must cool.
[] A shot only affects the square it hits.
[] Shots and grenades move 5 times faster than you do.
[] To stab someone, you must face that player and move at them.
[] Stabbing does 2 points worth of damage and shooting does 5 points.
[] Slime does 5 points of damage each time it hits.
[] You start with 15 charges and get 5 more for every new player.
[] A grenade affects the nine squares centered about the square it hits.
[] A satchel affects the twenty-five squares centered about the square it hits.
[] A bomb affects the forty-nine squares centered about the square it hits.
[] Slime affects all squares it oozes over (15 or 30 respectively).
[] One small mine and one large mine is placed in the maze for every new player. A mine has a 5% probability of tripping when you walk directly at it; 50% when going sideways on to it; 95% when backing up on to it. Tripping a mine costs you 5 points or 10 points respectively.

Defusing a mine is worth 1 charge or 9 charges respectively.
[] You cannot see behind you.
[] Scanning lasts for (20 times the number of players) turns. Scanning takes 1 ammo charge, so don't waste all your charges scanning.
[] Cloaking lasts for 20 turns.
[] Whenever you kill someone, you get 2 more damage capacity points and 2 damage points taken away.
[] Maximum typeahead is 5 characters.
[] A shot destroys normal (i.e., non-diagonal, non-door) walls.
[] Diagonal walls deflect shots and change orientation.
[] Doors disperse shots in random directions (up, down, left, right).
[] Diagonal walls and doors cannot be destroyed by direct shots but may be destroyed by an adjacent grenade explosion.
[] Slime goes around walls, not through them.
[] Walls regenerate, reappearing in the order they were destroyed. One percent of the regenerated walls will be diagonal walls or doors. When a wall is generated directly beneath a player, he is thrown in a random direction for a random period of time. When he lands, he sustains damage (up to 20 percent of the amount of damage he had before impact); that is, the less damage he had, the more nimble he is and therefore less likely to hurt himself on landing.

[] The environment variable HUNT is checked to get the player name. If you don't have this variable set, hunt will ask you what name you want to play under. If it is set, you may also set up a single character keyboard map, but then you have to enumerate the options:

e.g. setenv HUNT ``name=Sneaky,mapkey=zoFfGg1f2g3F4G''

sets the player name to Sneaky, and the maps z to o, F to f, G to g, 1 to f, 2 to g, 3 to F, and 4 to G. The mapkey option must be last.
[] It's a boring game if you're the only one playing.

Your score is the ratio of number of kills to number of times you entered the game and is only kept for the duration of a single session of hunt.

Hunt normally drives up the load average to be about (number_of_players + 0.5) greater than it would be without a hunt game executing. A limit of three players per host and nine players total is enforced by hunt.

FILES

/usr/games/lib/hunt.driver game coordinator

AUTHORS

Conrad Huang, Ken Arnold, and Greg Couch; University of California, San Francisco, Computer Graphics Lab

ACKNOWLEDGEMENTS

We thank Don Kneller, John Thomason, Eric Pettersen, and Scott Weiner for providing endless hours of play-testing to improve the character of the game. We hope their significant others will forgive them; we certainly don't.

BUGS

To keep up the pace, not everything is as realistic as possible.

There were some bugs in early releases of 4.2 BSD that hunt helped discover; hunt will crash your system if those bugs haven't been fixed.

NAME

mille - play Mille Bournes

SYNOPSIS

/usr/games/mille [ file ]

DESCRIPTION

Mille plays a two-handed game reminiscent of the Parker Brother's game of Mille Bournes with you. The rules are described below. If a file name is given on the command line, the game saved in that file is started.

When a game is started up, the bottom of the score window will contain a list of commands. They are:

P Pick a card from the deck. This card is placed in the `P' slot in your hand.

D Discard a card from your hand. To indicate which card, type the number of the card in the hand (or "P" for the just-picked card) followed by a <RETURN> or <SPACE>. The <RETURN or <SPACE> is required to allow recovery from typos which can be very expensive, like discarding safeties.

U Use a card. The card is again indicated by its number, followed by a <RETURN> or <SPACE>.

O Toggle ordering the hand. By default off, if turned on it will sort the cards in your hand appropriately. This is not recommended for the impatient on slow terminals.

Q Quit the game. This will ask for confirmation, just to be sure. Hitting <DELETE> (or <RUBOUT>) is equivalent.

S Save the game in a file. If the game was started from a file, you will be given an opportunity to save it on the same file. If you don't wish to, or you did not start from a file, you will be asked for the file name. If you type a <RETURN> without a name, the save will be terminated and the game resumed.

R Redraw the screen from scratch. The command ^L (control `L') will also work.

W Toggle window type. This switches the score window between the startup window (with all the command names) and the end-of-game window. Using the end-of-game window saves time by eliminating the switch at the end of the game to show the final score. Recommended for hackers and other miscreants.

If you make a mistake, an error message will be printed on the last line of the score window, and a bell will beep.

At the end of each hand or game, you will be asked if you wish to play another. If not, it will ask you if you want to save the game. If you do, and the save is unsuccessful, play will be resumed as if you had said you wanted to play another hand/game. This allows you to use the " .}S 3 1 "" " "S"" "" " " "" "" "" "" command to reattempt the save.

AUTHOR

Ken Arnold
(The game itself is a product of Parker Brothers, Inc.)

SEE ALSO

curses(3X), Screen Updating and Cursor Movement Optimization: A Library Package, Ken Arnold

CARDS

Here is some useful information. The number in parentheses after the card name is the number of that card in the deck:

| Hazard | Repair | Safety |
|---|---|---|
| Out of Gas (2) | Gasoline (6) | Extra Tank (1) |
| Flat Tire (2) | Spare Tire (6) | Puncture Proof (1) |
| Accident (2) | Repairs (6) | Driving Ace (1) |
| Stop (4) | Go (14) | Right of Way (1) |
| Speed Limit (3) | End of Limit (6) | |

25 - (10), 50 - (10), 75 - (10), 100 - (12), 200 - (4)

RULES

Object: The point of this game is to get a total of 5000 points in several hands. Each hand is a race to put down exactly 700 miles before your opponent does. Beyond the points gained by putting down milestones, there are several other ways of making points.

Overview: The game is played with a deck of 101 cards. Distance cards represent a number of miles traveled. They come in denominations of 25, 50, 75, 100, and 200. When one is played, it adds that many miles to the player's trip so far this hand. Hazard cards are used to prevent your opponent from putting down Distance cards. They can only be played if your opponent has a Go card on top of the Battle pile. The cards are Out of Gas, Accident, Flat Tire, Speed Limit, and Stop. Remedy cards fix problems caused by Hazard cards played on you by your opponent. The cards are Gasoline, Repairs, Spare Tire, End of Limit, and Go. Safety cards prevent your opponent from putting specific Hazard cards on

you in the first place. They are Extra Tank, Driving Ace, Puncture Proof, and Right of Way, and there are only one of each in the deck.

Board Layout: The board is split into several areas. From top to bottom, they are: SAFETY AREA (unlabeled): This is where the safeties will be placed as they are played. HAND: These are the cards in your hand. BATTLE: This is the Battle pile. All the Hazard and Remedy Cards are played here, except the Speed Limit and End of Limit cards. Only the top card is displayed, as it is the only effective one. SPEED: The Speed pile. The Speed Limit and End of Limit cards are played here to control the speed at which the player is allowed to put down miles. MILEAGE: Miles are placed here. The total of the numbers shown here is the distance traveled so far.

Play: The first pick alternates between the two players. Each turn usually starts with a pick from the deck. The player then plays a card, or if this is not possible or desirable, discards one. Normally, a play or discard of a single card constitutes a turn. If the card played is a safety, however, the same player takes another turn immediately.

This repeats until one of the players reaches 700 points or the deck runs out. If someone reaches 700, they have the option of going for an Extension, which means that the play continues until someone reaches 1000 miles.

Hazard and Remedy Cards: Hazard Cards are played on your opponent's Battle and Speed piles. Remedy Cards are used for undoing the effects of your opponent's nastiness.

Go (Green Light) must be the top card on your Battle pile for you to play any mileage, unless you have played the Right of Way card (see below).

Stop is played on your opponent's Go card to prevent them from playing mileage until they play a Go card.

Speed Limit is played on your opponent's Speed pile. Until they play an End of Limit they can only play 25 or 50 mile cards, presuming their Go card allows them to do even that.

End of Limit is played on your Speed pile to nullify a Speed Limit played by your opponent.

Out of Gas is played on your opponent's Go card. They must then play a Gasoline card, and then a Go card before they can play any more mileage.

Flat Tire is played on your opponent's Go card. They must then play a Spare Tire card, and then a Go card before they can play any more mileage.

Accident is played on your opponent's Go card. They

must then play a Repairs card, and then a Go card before they can play any more mileage.

Safety Cards: Safety cards prevent your opponent from playing the corresponding Hazard cards on you for the rest of the hand. It cancels an attack in progress, and always entitles the player to an extra turn.

Right of Way prevents your opponent from playing both Stop and Speed Limit cards on you. It also acts as a permanent Go card for the rest of the hand, so you can play mileage as long as there is not a Hazard card on top of your Battle pile. In this case only, your opponent can play Hazard cards directly on a Remedy card other than a Go card.

Extra Tank When played, your opponent cannot play an Out of Gas on your Battle Pile.

Puncture Proof When played, your opponent cannot play a Flat Tire on your Battle Pile.

Driving Ace When played, your opponent cannot play an Accident on your Battle Pile.

Distance Cards: Distance cards are played when you have a Go card on your Battle pile, or a Right of Way in your Safety area and are not stopped by a Hazard Card. They can be played in any combination that totals exactly 700 miles, except that you cannot play more than two 200 mile cards in one hand. A hand ends whenever one player gets exactly 700 miles or the deck runs out. In that case, play continues until neither someone reaches 700, or neither player can use any cards in their hand. If the trip is completed after the deck runs out, this is called Delayed Action.

Coup Fourr: This is a French fencing term for a counter-thrust move as part of a parry to an opponents attack. In Mille Bournes, it is used as follows: If an opponent plays a Hazard card, and you have the corresponding Safety in your hand, you play it immediately, even before you draw. This immediately removes the Hazard card from your Battle pile, and protects you from that card for the rest of the game. This gives you more points (see "Scoring" below).

Scoring: Scores are totaled at the end of each hand, whether or not anyone completed the trip. The terms used in the Score window have the following meanings:

Milestones Played: Each player scores as many miles as they played before the trip ended.

Each Safety: 100 points for each safety in the Safety area.

All 4 Safeties: 300 points if all four safeties are played.

Each Coup Four: 300 points for each Coup Foure accomplished.

The following bonus scores can apply only to the winning player.

Trip Completed: 400 points bonus for completing the trip to 700 or 1000.

Safe Trip: 300 points bonus for completing the trip without using any 200 mile cards.

Delayed Action: 300 points bonus for finishing after the deck was exhausted.

Extension: 200 points bonus for completing a 1000 mile trip.

Shut-Out: 500 points bonus for completing the trip before your opponent played any mileage cards.

Running totals are also kept for the current score for each player for the hand (Hand Total), the game (Overall Total), and number of games won (Games).

NAME

monop - Monopoly game

SYNOPSIS

/usr/games/monop [ file ]

DESCRIPTION

Monop is reminiscent of the Parker Brother's game Monopoly, and monitors a game between 1 to 9 users. It is assumed that the rules of Monopoly are known. The game follows the standard rules, with the exception that, if a property goes up for auction and there are only two solvent players, no auction is held and the property remains unowned.

The game, in effect, lends the player money, so it is possible to buy something which you cannot afford. However, as soon as a person goes into debt, he must "fix the problem", i.e., make himself solvent, before play can continue. If this is not possible, the player's property reverts to his debtee, either a player or the bank. A player can resign at any time to any person or the bank, which puts the property back on the board, unowned.

Any time that the response to a question is a string, e.g., a name, place or person, you can type `?' to get a list of valid answers. It is not possible to input a negative number, nor is it ever necessary.

A Summary of Commands:

quit: quit game: This allows you to quit the game. It asks you if you're sure.

print: print board: This prints out the current board. The columns have the following meanings (column headings are the same for the where, own holdings, and holdings commands):

Name The first ten characters of the name of the square

Own The number of the owner of the property.

Price The cost of the property (if any)

Mg This field has a `*' in it if the property is mortgaged

# If the property is a Utility or Railroad, this is the number of such owned by the owner. If the property is land, this is the number of houses on it.

Rent Current rent on the property. If it is not owned, there is no rent.

where: where players are: Tells you where all the players are. A `*' indicates the current player.

own holdings: List your own holdings, i.e., money, get-out-of-jail-free cards, and property.

holdings: holdings list: Look at anyone's holdings. It will ask you whose holdings you wish to look at. When you are finished, type "done".

shell: shell escape: Escape to a shell. When the shell dies, the program continues where you left off.

mortgage: mortgage property: Sets up a list of mortgageable property, and asks which you wish to mortgage.

unmortgage: unmortgage property: Unmortgage mortgaged property.

buy: buy houses: Sets up a list of monopolies on which you can buy houses. If there is more than one, it asks you which you want to buy for. It then asks you how many for each piece of property, giving the current amount in parentheses after the property name. If you build in an unbalanced manner (a disparity of more than one house within the same monopoly), it asks you to re-input things.

sell: sell houses: Sets up a list of monopolies from which you can sell houses. It operates in an analogous manner to buy.

card: card for jail: Use a get-out-of-jail-free card to get out of jail. If you're not in jail, or you don't have one, it tells you so.

pay: pay for jail: Pay $50 to get out of jail, from whence you are put on Just Visiting. Difficult to do if you're not there.

trade: This allows you to trade with another player. It asks you whom you wish to trade with, and then asks you what each wishes to give up. You can get a summary at the end, and, in all cases, it asks for confirmation of the trade before doing it.

resign: Resign to another player or the bank. If you resign to the bank, all property reverts to its virgin state, and get-out-of-jail free cards revert to the deck.

save: save game: Save the current game in a file for later play. You can continue play after saving, either by adding the file in which you saved the game after the monop command, or by using the restore command (see below). It will ask you which file you wish to save it in, and, if the file exists, confirm that you wish to overwrite it.

restore: restore game: Read in a previously saved game from a file. It leaves the file intact.

roll: Roll the dice and move forward to your new location. If you simply hit the <RETURN> key instead of a command, it is the same as typing roll.

AUTHOR

Ken Arnold

FILES

/usr/games/lib/cards.pck Chance and Community Chest cards

BUGS

No command can be given an argument instead of a response to a query.

NAME

moo - guessing game

SYNOPSIS

/usr/games/moo

DESCRIPTION

Moo is a guessing game imported from England. The computer picks a number consisting of four distinct decimal digits. The player guesses four distinct digits being scored on each guess. A `cow' is a correct digit in an incorrect position. A `bull' is a correct digit in a correct position. The game continues until the player guesses the number (a score of four bulls).

NAME

number - convert Arabic numerals to English

SYNOPSIS

/usr/games/number

DESCRIPTION

Number copies the standard input to the standard output, changing each decimal number to a fully spelled out version.

NAME

quiz - test your knowledge

SYNOPSIS

/usr/games/quiz [ -i file ] [ -t ] [ category1 category2 ]

DESCRIPTION

Quiz gives associative knowledge tests on various subjects. It asks items chosen from category1 and expects answers from category2. If no categories are specified, quiz gives instructions and lists the available categories.

Quiz tells a correct answer whenever you type a bare newline. At the end of input, upon interrupt, or when questions run out, quiz reports a score and terminates.

The -t flag specifies `tutorial' mode, where missed questions are repeated later, and material is gradually introduced as you learn.

The -i flag causes the named file to be substituted for the default index file. The lines of these files have the syntax:

```
line      = category newline | category `:' line
category  = alternate | category `|' alternate
alternate = empty | alternate primary
primary   = character | `[' category `]' | option
option    = `{' category `}'
```

The first category on each line of an index file names an information file. The remaining categories specify the order and contents of the data in each line of the information file. Information files have the same syntax. Backslash `\' is used as with sh(1) to quote syntactically significant characters or to insert transparent newlines into a line. When either a question or its answer is empty, quiz will refrain from asking it.

FILES

/usr/games/quiz.k/*

BUGS

The construct `a|ab' doesn't work in an information file. Use `a{b}'.

NAME

rain - animated raindrops display

SYNOPSIS

/usr/games/rain

DESCRIPTION

Rain's display is modeled after the VAX/VMS program of the same name. The terminal has to be set for 9600 baud to obtain the proper effect.

As with all programs that use termcap, the TERM environment variable must be set (and exported) to the type of the terminal being used.

FILES

/etc/termcap

AUTHOR

Eric P. Scott

NAME

robots - fight off villainous robots

SYNOPSIS

/usr/games/robots [ -sjta ] [ scorefile ]

DESCRIPTION

Robots pits you against evil robots, who are trying to kill you (which is why they are evil). Fortunately for you, even though they are evil, they are not very bright and have a habit of bumping into each other, thus destroying themselves. In order to survive, you must get them to kill each other off, since you have no offensive weaponry.

Since you are stuck without offensive weaponry, you are endowed with one piece of defensive weaponry: a teleportation device. When two robots run into each other or a junk pile, they die. If a robot runs into you, you die. When a robot dies, you get 10 points, and when all the robots die, you start on the next field. This keeps up until they finally get you.

Robots are represented on the screen by a `+', the junk heaps from their collisions by a `*', and you (the good guy) by a `@'.

The commands are:

| | |
|---|---|
| h | move one square left |
| l | move one square right |
| k | move one square up |
| j | move one square down |
| y | move one square up and left |
| u | move one square up and right |
| b | move one square down and left |
| n | move one square down and right |
| . | (also space) do nothing for one turn |
| HJKLBNYU | run as far as possible in the given direction |
| > | do nothing for as long as possible |
| t | teleport to a random location |
| w | wait until you die or they all do |
| q | quit |
| ^L | redraw the screen |

All commands can be preceded by a count.

If you use the `w' command and survive to the next level, you will get a bonus of 10% for each robot which died after you decided to wait. If you die, however, you get nothing. For all other commands, the program will save you from typos by stopping short of being eaten. However, with `w' you take the risk of dying by miscalculation.

Only five scores are allowed per user on the score file. If you make it into the score file, you will be shown the list at the end of the game. If an alternate score file is specified, that will be used instead of the standard file for scores.

The options are

-s Don't play, just show the score file

-j Jump, i.e., when you run, don't show any intermediate positions; only show things at the end. This is useful on slow terminals.

-t Teleport automatically when you have no other option. This is a little disconcerting until you get used to it, and then it is very nice.

-a Advance into the higher levels directly, skipping the lower, easier levels.

AUTHOR

Ken Arnold

FILES

/usr/games/lib/robots_roll the score file

BUGS

Bugs? You crazy, man?!?

NAME

rogue - Exploring The Dungeons of Doom

SYNOPSIS

/usr/games/rogue [ -r ] [ save_file ] [ -s ] [ -d ]

DESCRIPTION

Rogue is a computer fantasy game with a new twist. It is crt oriented and the object of the game is to survive the attacks of various monsters and get a lot of gold, rather than the puzzle solving orientation of most computer fantasy games.

To get started you really only need to know two commands. The command ? will give you a list of the available commands and the command / will identify the things you see on the screen.

To win the game (as opposed to merely playing to beat other people's high scores) you must locate the Amulet of Yendor which is somewhere below the 20th level of the dungeon and get it out. Nobody has achieved this yet and if somebody does, they will probably go down in history as a hero among heroes.

When the game ends, either by your death, when you quit, or if you (by some miracle) manage to win, rogue will give you a list of the top-ten scorers. The scoring is based entirely upon how much gold you get. There is a 10% penalty for getting yourself killed.

If save_file is specified, rogue will be restored from the specified saved game file. If the -r option is used, the save game file is presumed to be the default.

The -s option will print out the list of scores.

The -d option will kill you and try to add you to the score file.

For more detailed directions, read the document A Guide to the Dungeons of Doom.

AUTHORS

Michael C. Toy, Kenneth C. R. C. Arnold, Glenn Wichman

FILES

/usr/games/lib/rogue_roll Score file
~/rogue.save Default save file

SEE ALSO

Michael C. Toy and Kenneth C. R. C. Arnold, A guide to the

Dungeons of Doom

BUGS

Probably infinite (although countably infinite). However, that Ice Monsters sometimes transfix you permanently is not a bug. It's a feature.

NAME

sail - multi-user wooden ships and iron men

SYNOPSIS

sail [ -s [ -l ] ] [ -x ] [ -b ] [ num ]

DESCRIPTION

Sail is a computer version of Avalon Hill's game of fighting sail originally developed by S. Craig Taylor.

Players of Sail take command of an old fashioned Man of War and fight other players or the computer. They may re-enact one of the many historical sea battles recorded in the game, or they can choose a fictional battle.

As a sea captain in the Sail Navy, the player has complete control over the workings of his ship. He must order every maneuver, change the set of his sails, and judge the right moment to let loose the terrible destruction of his broadsides. In addition to fighting the enemy, he must harness the powers of the wind and sea to make them work for him. The outcome of many battles during the age of sail was decided by the ability of one captain to hold the `weather gage.'

The flags are:

-s Print the names and ships of the top ten sailors.

-l Show the login name. Only effective with -s.

-x Play the first available ship instead of prompting for a choice.

-b No bells.

IMPLEMENTATION

Sail is really two programs in one. Each player starts up a process which runs his own ship. In addition, a driver process is forked (by the first player) to run the computer ships and take care of global bookkeeping.

Because the driver must calculate moves for each ship it controls, the more ships the computer is playing, the slower the game will appear.

If a player joins a game in progress, he will synchronize with the other players (a rather slow process for everyone), and then he may play along with the rest.

To implement a multi-user game in Version 7 UNIX, which was the operating system Sail was first written under, the

communicating processes must use a common temporary file as a place to read and write messages. In addition, a locking mechanism must be provided to ensure exclusive access to the shared file. For example, Sail uses a temporary file named /tmp/#sailsink.21 for scenario 21, and corresponding file names for the other scenarios. To provide exclusive access to the temporary file, Sail uses a technique stolen from an old game called "pubcaves" by Jeff Cohen. Processes do a busy wait in the loop

```
for (n = 0; link(sync_file, sync_lock) < 0 && n < 30; n++)
                    sleep(2);
```

until they are able to create a link to a file named "/tmp/#saillock.??". The "??" correspond to the scenario number of the game. Since UNIX guarantees that a link will point to only one file, the process that succeeds in linking will have exclusive access to the temporary file.

Whether or not this really works is open to speculation. When ucbmiro was rebooted after a crash, the file system check program found 3 links between the Sail temporary file and its link file.

CONSEQUENCES OF SEPARATE PLAYER AND DRIVER

When players do something of global interest, such as moving or firing, the driver must coordinate the action with the other ships in the game. For example, if a player wants to move in a certain direction, he writes a message into the temporary file requesting the driver to move his ship. Each ``turn,'' the driver reads all the messages sent from the players and decides what happened. It then writes back into the temporary file new values of variables, etc.

The most noticeable effect this communication has on the game is the delay in moving. Suppose a player types a move for his ship and hits return. What happens then? The player process saves up messages to be written to the temporary file in a buffer. Every 7 seconds or so, the player process gets exclusive access to the temporary file and writes out its buffer to the file. The driver, running asynchronously, must read in the movement command, process it, and write out the results. This takes two exclusive accesses to the temporary file. Finally, when the player process gets around to doing another 7 second update, the results of the move are displayed on the screen. Hence, every movement requires four exclusive accesses to the temporary file (anywhere from 7 to 21 seconds depending upon asynchrony) before the player sees the results of his moves.

In practice, the delays are not as annoying as they would appear. There is room for "pipelining" in the movement.

After the player writes out a first movement message, a second movement command can then be issued. The first message will be in the temporary file waiting for the driver, and the second will be in the file buffer waiting to be written to the file. Thus, by always typing moves a turn ahead of the time, the player can sail around quite quickly.

If the player types several movement commands between two 7 second updates, only the last movement command typed will be seen by the driver. Movement commands within the same update "overwrite" each other, in a sense.

## THE HISTORY OF SAIL

I wrote the first version of Sail on a PDP 11/70 in the fall of 1980. Needless to say, the code was horrendous, not portable in any sense of the word, and didn't work. The program was not very modular and had fseeks() and fwrites() every few lines. After a tremendous rewrite from the top down, I got the first working version up by 1981. There were several annoying bugs concerning firing broadsides and finding angles. Sail uses no floating point, by the way, so the direction routines are rather tricky. Ed Wang rewrote my angle() routine in 1981 to be more correct (although it still doesn't work perfectly), and he added code to let a player select which ship he wanted at the start of the game (instead of the first one available).

Captain Happy (Craig Leres) is responsible for making Sail portable for the first time. This was no easy task, by the way. Constants like 2 and 10 were very frequent in the code. I also became famous for using "Riggle Memorial Structures" in Sail. Many of my structure references are so long that they run off the line printer page. Here is an example, if you promise not to laugh.

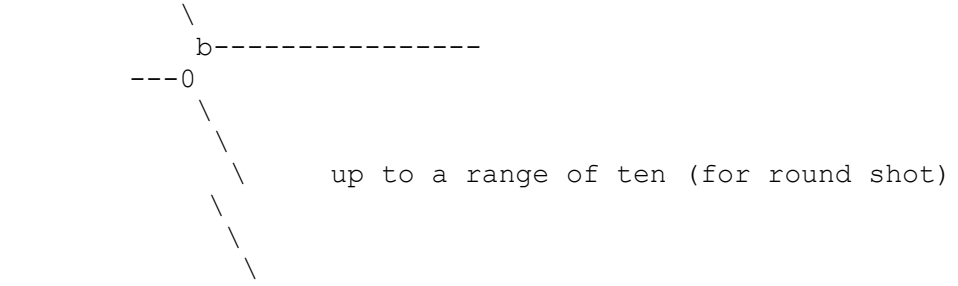
```
specs[scene[flog.fgamenum].ship[flog.fshipnum].shipnum].pts
```

Sail received its fourth and most thorough rewrite in the summer and fall of 1983. Ed Wang rewrote and modularized the code (a monumental feat) almost from scratch. Although he introduced many new bugs, the final result was very much cleaner and (?) faster. He added window movement commands and find ship commands.

## HISTORICAL INFO

Old Square Riggers were very maneuverable ships capable of intricate sailing. Their only disadvantage was an inability to sail very close to the wind. The design of a wooden ship allowed only for the guns to bear to the left and right sides. A few guns of small aspect (usually 6 or 9 pounders) could point forward, but their effect was small compared to

a 68 gun broadside of 24 or 32 pounders. The guns bear approximately like so:




An interesting phenomenon occurred when a broadside was fired down the length of an enemy ship. The shot tended to bounce along the deck and did several times more damage. This phenomenon was called a rake. Because the bows of a ship are very strong and present a smaller target than the stern, a stern rake (firing from the stern to the bow) causes more damage than a bow rake.


Most ships were equipped with carronades, which were very large, close range cannons. American ships from the revolution until the War of 1812 were almost entirely armed with carronades.

The period of history covered in Sail is approximately from the 1770's until the end of Napoleanic France in 1815. There are many excellent books about the age of sail. My favorite author is Captain Frederick Marryat. More contemporary authors include C.S. Forester and Alexander Kent.

Fighting ships came in several sizes classed by armament. The mainstays of any fleet were its "Ships of the Line", or "Line of Battle Ships". They were so named because these ships fought together in great lines. They were close enough for mutual support, yet every ship could fire both its broadsides. We get the modern words "ocean liner," or "liner," and "battleship" from "ship of the line." The most common size was the the 74 gun two decked ship of the line. The two gun decks usually mounted 18 and 24 pounder guns.

The pride of the fleet were the first rates. These were huge three decked ships of the line mounting 80 to 136 guns. The guns in the three tiers were usually 18, 24, and 32 pounders in that order from top to bottom.

Various other ships came next. They were almost all "razees," or ships of the line with one deck sawed off. They mounted 40-64 guns and were a poor cross between a frigate and a line of battle ship. They neither had the speed of the former nor the firepower of the latter.

Next came the "eyes of the fleet." Frigates came in many sizes mounting anywhere from 32 to 44 guns. They were very handy vessels. They could outsail anything bigger and outshoot anything smaller. Frigates didn't fight in lines of battle as the much bigger 74's did. Instead, they harassed the enemy's rear or captured crippled ships. They were much more useful in missions away from the fleet, such as cutting out expeditions or boat actions. They could hit hard and get away fast.

Lastly, there were the corvettes, sloops, and brigs. These were smaller ships mounting typically fewer than 20 guns. A corvette was only slightly smaller than a frigate, so one might have up to 30 guns. Sloops were used for carrying dispatches or passengers. Brigs were something you built for land-locked lakes.

SAIL PARTICULARS

Ships in Sail are represented by two characters. One character represents the bow of the ship, and the other represents the stern. Ships have nationalities and numbers. The first ship of a nationality is number 0, the second number 1, etc. Therefore, the first British ship in a game would be printed as "b0". The second Brit would be "b1", and the fifth Don would be "s4".

Ships can set normal sails, called Battle Sails, or bend on extra canvas called Full Sails. A ship under full sail is a beautiful sight indeed, and it can move much faster than a ship under Battle Sails. The only trouble is, with full sails set, there is so much tension on sail and rigging that a well aimed round shot can burst a sail into ribbons where it would only cause a little hole in a loose sail. For this reason, rigging damage is doubled on a ship with full sails set. Don't let that discourage you from using full sails. I like to keep them up right into the heat of battle. A ship with full sails set has a capital letter for its nationality. E.g., a Frog, "f0", with full sails set would be printed as "F0".

When a ship is battered into a listing hulk, the last man aboard "strikes the colors." This ceremony is the ship's formal surrender. The nationality character of a surrendered ship is printed as "!". E.g., the Frog of our last example would soon be "!0".

A ship has a random chance of catching fire or sinking when it reaches the stage of listing hulk. A sinking ship has a "~" printed for its nationality, and a ship on fire and about to explode has a "#" printed.

Captured ships become the nationality of the prize crew. Therefore, if an American ship captures a British ship, the British ship will have an "a" printed for its nationality. In addition, the ship number is changed to "&","'", "(", ,")", "*", or "+" depending upon the original number, be it 0,1,2,3,4, or 5. E.g., the "b0" captured by an American becomes the "a&". The "s4" captured by a Frog becomes the "f*".

The ultimate example is, of course, an exploding Brit captured by an American: "#&".

MOVEMENT

Movement is the most confusing part of Sail to many. Ships can head in 8 directions:

The stern of a ship moves when it turns. The bow remains stationary. Ships can always turn, regardless of the wind (unless they are becalmed). All ships drift when they lose headway. If a ship doesn't move forward at all for two turns, it will begin to drift. If a ship has begun to drift, then it must move forward before it turns, if it plans to do more than make a right or left turn, which is always possible.

Movement commands to Sail are a string of forward moves and turns. An example is "l3". It will turn a ship left and then move it ahead 3 spaces. In the drawing above, the "b0" made 7 successive left turns. When Sail prompts you for a move, it prints three characters of import. E.g.,

```
   move (7, 4):
```

The first number is the maximum number of moves you can make, including turns. The second number is the maximum number of turns you can make. Between the numbers is sometimes printed a quote "'". If the quote is present, it means that your ship has been drifting, and you must move ahead to regain headway before you turn (see note above). Some of the possible moves for the example above are as follows:

```
   move (7, 4): 7
   move (7, 4): 1
   move (7, 4): d      /* drift, or do nothing */
```

```
move (7, 4): 6r
move (7, 4): 5r1
move (7, 4): 4r1r
move (7, 4): l1r1r2
move (7, 4): 1r1r1r1
```

Because square riggers performed so poorly sailing into the wind, if at any point in a movement command you turn into the wind, the movement stops there. E.g.,

```
move (7, 4): l1l4
Movement Error;
Helm: l1l
```

Moreover, whenever you make a turn, your movement allowance drops to min(what's left, what you would have at the new attitude). In short, if you turn closer to the wind, you most likely won't be able to sail the full allowance printed in the "move" prompt.

Old sailing captains had to keep an eye constantly on the wind. Captains in Sail are no different. A ship's ability to move depends on its attitide to the wind. The best angle possible is to have the wind off your quarter, that is, just off the stern. The direction rose on the side of the screen gives the possible movements for your ship at all positions to the wind. Battle sail speeds are given first, and full sail speeds are given in parenthesis.

```
 0 1(2)
\|/
-^-3(6)
/|\
 | 4(7)
3(6)
```

Pretend the bow of your ship (the "^") is pointing upward and the wind is blowing from the bottom to the top of the page. The numbers at the bottom "3(6)" will be your speed under battle or full sails in such a situation. If the wind is off your quarter, then you can move "4(7)". If the wind is off your beam, "3(6)". If the wind is off your bow, then you can only move "1(2)". Facing into the wind, you can't move at all. Ships facing into the wind were said to be "in irons".

WINDSPEED AND DIRECTION

The windspeed and direction is displayed as a little weather vane on the side of the screen. The number in the middle of the vane indicates the wind speed, and the + to - indicates the wind direction. The wind blows from the + sign (high pressure) to the - sign (low pressure). E.g.,

```
|
3
+
```

The wind speeds are 0 = becalmed, 1 = light breeze, 2 = moderate breeze, 3 = fresh breeze, 4 = strong breeze, 5 = gale, 6 = full gale, 7 = hurricane. If a hurricane shows up, all ships are destroyed.

GRAPPLING AND FOULING

If two ships collide, they run the risk of becoming tangled together. This is called "fouling." Fouled ships are stuck together, and neither can move. They can unfoul each other if they want to. Boarding parties can only be sent across to ships when the antagonists are either fouled or grappled.

Ships can grapple each other by throwing grapnels into the rigging of the other.

The number of fouls and grapples you have are displayed on the upper right of the screen.

BOARDING

Boarding was a very costly venture in terms of human life. Boarding parties may be formed in Sail to either board an enemy ship or to defend your own ship against attack. Men organized as Defensive Boarding Parties fight twice as hard to save their ship as men left unorganized.

The boarding strength of a crew depends upon its quality and upon the number of men sent.

CREW QUALITY

The British seaman was world renowned for his sailing abilities. American sailors, however, were actually the best seamen in the world. Because the American Navy offered twice the wages of the Royal Navy, British seamen who liked the sea defected to America by the thousands.

In Sail, crew quality is quantized into 5 energy levels. "Elite" crews can outshoot and outfight all other sailors. "Crack" crews are next. "Mundane" crews are average, and "Green" and "Mutinous" crews are below average. A good rule of thumb is that "Crack" or "Elite" crews get one extra hit per broadside compared to "Mundane" crews. Don't expect too much from "Green" crews.

BROADSIDES

Your two broadsides may be loaded with four kinds of shot: grape, chain, round, and double. You have guns and carronades in both the port and starboard batteries.

Carronades only have a range of two, so you have to get in close to be able to fire them. You have the choice of firing at the hull or rigging of another ship. If the range of the ship is greater than 6, then you may only shoot at the rigging.

The types of shot and their advantages are:

ROUND
Range of 10. Good for hull or rigging hits.

DOUBLE
Range of 1. Extra good for hull or rigging hits. Double takes two turns to load.

CHAIN
Range of 3. Excellent for tearing down rigging. Cannot damage hull or guns, though.

GRAPE
Range of 1. Sometimes devastating against enemy crews.

On the side of the screen is displayed some vital information about your ship:

```
Load  D! R!
Hull  9
Crew  4  4   2
Guns  4  4
Carr  2  2
Rigg  5 5 5 5
```

"Load" shows what your port (left) and starboard (right) broadsides are loaded with. A "!" after the type of shot indicates that it is an initial broadside. Initial broadside were loaded with care before battle and before the decks ran red with blood. As a consequence, initial broadsides are a little more effective than broadsides loaded later. A "*" after the type of shot indicates that the gun crews are still loading it, and you cannot fire yet. "Hull" shows how much hull you have left. "Crew" shows your three sections of crew. As your crew dies off, your ability to fire decreases. "Guns" and "Carr" show your port and starboard guns. As you lose guns, your ability to fire decreases. "Rigg" shows how much rigging you have on your 3 or 4 masts. As rigging is shot away, you lose mobility.

EFFECTIVENESS OF FIRE
It is very dramatic when a ship fires its thunderous broadsides, but the mere opportunity to fire them does not guarantee any hits. Many factors influence the destructive force of a broadside. First of all, and the chief factor,

is distance. It is harder to hit a ship at range ten than it is to hit one sloshing alongside. Next is raking. Raking fire, as mentioned before, can sometimes dismast a ship at range ten. Next, crew size and quality affects the damage done by a broadside. The number of guns firing also bears on the point, so to speak. Lastly, weather affects the accuracy of a broadside. If the seas are high (5 or 6), then the lower gunports of ships of the line can't even be opened to run out the guns. This gives frigates and other flush decked vessels an advantage in a storm. The scenario Pellew vs. The Droits de L'Homme takes advantage of this peculiar circumstance.

REPAIRS

Repairs may be made to your Hull, Guns, and Rigging at the slow rate of two points per three turns. The message "Repairs Completed" will be printed if no more repairs can be made.

PECULIARITIES OF COMPUTER SHIPS

Computer ships in Sail follow all the rules above with a few exceptions. Computer ships never repair damage. If they did, the players could never beat them. They play well enough as it is. As a consolation, the computer ships can fire double shot every turn. That fluke is a good reason to keep your distance. The Driver figures out the moves of the computer ships. It computes them with a typical A.I. distance function and a depth first search to find the maximum "score." It seems to work fairly well, although I'll be the first to admit it isn't perfect.

HOW TO PLAY

Commands are given to Sail by typing a single character. You will then be prompted for further input. A brief summary of the commands follows.

COMMAND SUMMARY

| | |
|---|---|
| 'f' | Fire broadsides if they bear |
| 'l' | Reload |
| 'L' | Unload broadsides (to change ammo) |
| 'm' | Move |
| 'i' | Print the closest ship |
| 'I' | Print all ships |
| 'F' | Find a particular ship or ships (e.g. "a?" for all Americans) |
| 's' | Send a message around the fleet |
| 'b' | Attempt to board an enemy ship |
| 'B' | Recall boarding parties |
| 'c' | Change set of sail |
| 'r' | Repair |
| 'u' | Attempt to unfoul |
| 'g' | Grapple/ungrapple |
| 'v' | Print version number of game |
| '^L' | Redraw screen |
| 'Q' | Quit |

| | |
|---|---|
| 'C' | Center your ship in the window |
| 'U' | Move window up |
| 'D','N' | Move window down |
| 'H' | Move window left |
| 'J' | Move window right |
| 'S' | Toggle window to follow your ship or stay where it is |

SCENARIOS

Here is a summary of the scenarios in Sail:

Ranger vs. Drake:

Wind from the N, blowing a fresh breeze.

(a) Ranger 19 gun Sloop (crack crew) (7 pts)
(b) Drake 17 gun Sloop (crack crew) (6 pts)

The Battle of Flamborough Head:

Wind from the S, blowing a fresh breeze.

This is John Paul Jones' first famous battle. Aboard the Bonhomme Richard, he was able to overcome the Serapis's greater firepower by quickly boarding her.

(a) Bonhomme Rich 42 gun Corvette (crack crew) (11 pts)
(b) Serapis 44 gun Frigate (crack crew) (12 pts)

Arbuthnot and Des Touches:

Wind from the N, blowing a gale.

(b) America 64 gun Ship of the Line (crack crew) (20 pts)
(b) Befford 74 gun Ship of the Line (crack crew) (26 pts)

```
     (b) Adamant       50 gun Ship of the Line (crack crew) (17 pts)
     (b) London        98 gun 3 Decker SOL (crack crew) (28 pts)
     (b) Royal Oak     74 gun Ship of the Line (crack crew) (26 pts)
     (f) Neptune       74 gun Ship of the Line (average crew) (24 pts)
     (f) Duc Bougogne  80 gun 3 Decker SOL (average crew) (27 pts)
     (f) Conquerant    74 gun Ship of the Line (average crew) (24 pts)
     (f) Provence      64 gun Ship of the Line (average crew) (18 pts)
     (f) Romulus       44 gun Ship of the Line (average crew) (10 pts)

Suffren and Hughes:
     Wind from the S, blowing a fresh breeze.

     (b) Monmouth      74 gun Ship of the Line (average crew) (24 pts)
     (b) Hero          74 gun Ship of the Line (crack crew) (26 pts)
     (b) Isis          50 gun Ship of the Line (crack crew) (17 pts)
     (b) Superb        74 gun Ship of the Line (crack crew) (27 pts)
     (b) Burford       74 gun Ship of the Line (average crew) (24 pts)
     (f) Flamband      50 gun Ship of the Line (average crew) (14 pts)
     (f) Annibal       74 gun Ship of the Line (average crew) (24 pts)
     (f) Severe        64 gun Ship of the Line (average crew) (18 pts)
     (f) Brilliant     80 gun Ship of the Line (crack crew) (31 pts)
     (f) Sphinx        80 gun Ship of the Line (average crew) (27 pts)

Nymphe vs. Cleopatre:
     Wind from the S, blowing a fresh breeze.

     (b) Nymphe        36 gun Frigate (crack crew) (11 pts)
     (f) Cleopatre     36 gun Frigate (average crew) (10 pts)

Mars vs. Hercule:
     Wind from the S, blowing a fresh breeze.
     (b) Mars          74 gun Ship of the Line (crack crew) (26 pts)
     (f) Hercule       74 gun Ship of the Line (average crew) (23 pts)

Ambuscade vs. Baionnaise:
     Wind from the N, blowing a fresh breeze.

     (b) Ambuscade     32 gun Frigate (average crew) (9 pts)
     (f) Baionnaise    24 gun Corvette (average crew) (9 pts)

Constellation vs. Insurgent:
     Wind from the S, blowing a gale.

     (a) Constellation 38 gun Corvette (elite crew) (17 pts)
     (f) Insurgent     36 gun Corvette (average crew) (11 pts)

Constellation vs. Vengeance:
     Wind from the S, blowing a fresh breeze.

     (a) Constellation 38 gun Corvette (elite crew) (17 pts)
     (f) Vengeance     40 gun Frigate (average crew) (15 pts)
```

The Battle of Lissa:
Wind from the S, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (b) | Amphion | 32 gun Frigate | (elite crew) (13 pts) |
| (b) | Active | 38 gun Frigate | (elite crew) (18 pts) |
| (b) | Volage | 22 gun Frigate | (elite crew) (11 pts) |
| (b) | Cerberus | 32 gun Frigate | (elite crew) (13 pts) |
| (f) | Favorite | 40 gun Frigate | (average crew) (15 pts) |
| (f) | Flore | 40 gun Frigate | (average crew) (15 pts) |
| (f) | Danae | 40 gun Frigate | (crack crew) (17 pts) |
| (f) | Bellona | 32 gun Frigate | (green crew) (9 pts) |
| (f) | Corona | 40 gun Frigate | (green crew) (12 pts) |
| (f) | Carolina | 32 gun Frigate | (green crew) (7 pts) |

Constitution vs. Guerriere:
Wind from the SW, blowing a gale.

| | | | |
|---|---|---|---|
| (a) | Constitution | 44 gun Corvette | (elite crew) (24 pts) |
| (b) | Guerriere | 38 gun Frigate | (crack crew) (15 pts) |

United States vs. Macedonian:
Wind from the S, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (a) | United States | 44 gun Frigate | (elite crew) (24 pts) |
| (b) | Macedonian | 38 gun Frigate | (crack crew) (16 pts) |

Constitution vs. Java:
Wind from the S, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (a) | Constitution | 44 gun Corvette | (elite crew) (24 pts) |
| (b) | Java | 38 gun Corvette | (crack crew) (19 pts) |

Chesapeake vs. Shannon:
Wind from the S, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (a) | Chesapeake | 38 gun Frigate | (average crew) (14 pts) |
| (b) | Shannon | 38 gun Frigate | (elite crew) (17 pts) |

The Battle of Lake Erie:
Wind from the S, blowing a light breeze.

| | | | |
|---|---|---|---|
| (a) | Lawrence | 20 gun Sloop | (crack crew) (9 pts) |
| (a) | Niagara | 20 gun Sloop | (elite crew) (12 pts) |
| (b) | Lady Prevost | 13 gun Brig | (crack crew) (5 pts) |
| (b) | Detroit | 19 gun Sloop | (crack crew) (7 pts) |
| (b) | Q. Charlotte | 17 gun Sloop | (crack crew) (6 pts) |

Wasp vs. Reindeer:
Wind from the S, blowing a light breeze.

| | | | |
|---|---|---|---|
| (a) | Wasp | 20 gun Sloop | (elite crew) (12 pts) |
| (b) | Reindeer | 18 gun Sloop | (elite crew) (9 pts) |

Constitution vs. Cyane and Levant:
Wind from the S, blowing a moderate breeze.

| | | | |
|---|---|---|---|
| (a) | Constitution | 44 gun | Corvette (elite crew) (24 pts) |
| (b) | Cyane | 24 gun | Sloop (crack crew) (11 pts) (b) |
| | Levant | 20 gun | Sloop (crack crew) (10 pts) |

Pellew vs. Droits de L'Homme:
Wind from the N, blowing a gale.

| | | | |
|---|---|---|---|
| (b) | Indefatigable | 44 gun | Frigate (elite crew) (14 pts) |
| (b) | Amazon | 36 gun | Frigate (crack crew) (14 pts) |
| (f) | Droits L'Hom | 74 gun | Ship of the Line (average crew) (24 pts) |

Algeciras:
Wind from the SW, blowing a moderate breeze.

| | | | |
|---|---|---|---|
| (b) | Caesar | 80 gun | Ship of the Line (crack crew) (31 pts) |
| (b) | Pompee | 74 gun | Ship of the Line (crack crew) (27 pts) |
| (b) | Spencer | 74 gun | Ship of the Line (crack crew) (26 pts) |
| (b) | Hannibal | 98 gun | 3 Decker SOL (crack crew) (28 pts) |
| (s) | Real-Carlos | 112 gun | 3 Decker SOL (green crew) (27 pts) |
| (s) | San Fernando | 96 gun | 3 Decker SOL (green crew) (24 pts) |
| (s) | Argonauta | 80 gun | Ship of the Line (green crew) (23 pts) |
| (s) | San Augustine | 74 gun | Ship of the Line (green crew) (20 pts) |
| (f) | Indomptable | 80 gun | Ship of the Line (average crew) (27 pts) |
| (f) | Desaix | 74 gun | Ship of the Line (average crew) (24 pts) |

Lake Champlain:
Wind from the N, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (a) | Saratoga | 26 gun | Sloop (crack crew) (12 pts) |
| (a) | Eagle | 20 gun | Sloop (crack crew) (11 pts) |
| (a) | Ticonderoga | 17 gun | Sloop (crack crew) (9 pts) |
| (a) | Preble | 7 gun | Brig (crack crew) (4 pts) |
| (b) | Confiance | 37 gun | Frigate (crack crew) (14 pts) |
| (b) | Linnet | 16 gun | Sloop (elite crew) (10 pts) |
| (b) | Chubb | 11 gun | Brig (crack crew) (5 pts) |

Last Voyage of the USS President:
Wind from the N, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (a) | President | 44 gun | Frigate (elite crew) (24 pts) |
| (b) | Endymion | 40 gun | Frigate (crack crew) (17 pts) |
| (b) | Pomone | 44 gun | Frigate (crack crew) (20 pts) |
| (b) | Tenedos | 38 gun | Frigate (crack crew) (15 pts) |

Hornblower and the Natividad:
Wind from the E, blowing a gale.

A scenario for you Horny fans. Remember, he sank the Natividad against heavy odds and winds. Hint: don't try to

board the Natividad, her crew is much bigger, albeit green.

(b) Lydia 36 gun Frigate (elite crew) (13 pts)
(s) Natividad 50 gun Ship of the Line (green crew) (14 pts)

Curse of the Flying Dutchman:
Wind from the S, blowing a fresh breeze.

Just for fun, take the Piece of cake.

(s) Piece of Cake 24 gun Corvette (average crew) (9 pts)
(f) Flying Dutchy 120 gun 3 Decker SOL (elite crew) (43 pts)

The South Pacific:
Wind from the S, blowing a strong breeze.

(a) USS Scurvy 136 gun 3 Decker SOL (mutinous crew) (27 pts)
(b) HMS Tahiti 120 gun 3 Decker SOL (elite crew) (43 pts)
(s) Australian 32 gun Frigate (average crew) (9 pts)
(f) Bikini Atoll 7 gun Brig (crack crew) (4 pts)

Hornblower and the battle of Rosas
Wind from the E, blowing a fresh breeze.

The only battle Hornblower ever lost. He was able to dismast one ship and stern rake the others though. See if you can do as well.

(b) Sutherland 74 gun Ship of the Line (crack crew) (26 pts)
(f) Turenne 80 gun 3 Decker SOL (average crew) (27 pts)
(f) Nightmare 74 gun Ship of the Line (average crew) (24 pts)
(f) Paris 112 gun 3 Decker SOL (green crew) (27 pts)
(f) Napolean 74 gun Ship of the Line (green crew) (20 pts)

Cape Horn:
Wind from the NE, blowing a strong breeze.

(a) Concord 80 gun Ship of the Line (average crew) (27 pts)
(a) Berkeley 98 gun 3 Decker SOL (crack crew) (28 pts)
(b) Thames 120 gun 3 Decker SOL (elite crew) (43 pts)
(s) Madrid 112 gun 3 Decker SOL (green crew) (27 pts)
(f) Musket 80 gun 3 Decker SOL (average crew) (27 pts)

New Orleans:
Wind from the SE, blowing a fresh breeze.

Watch that little Cypress go!

(a) Alligator 120 gun 3 Decker SOL (elite crew) (43 pts)
(b) Firefly 74 gun Ship of the Line (crack crew) (27 pts)
(b) Cypress 44 gun Frigate (elite crew) (14 pts)

Botany Bay:
Wind from the N, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (b) | Shark | 64 gun Ship of the Line | (average crew) (18 pts) |
| (f) | Coral Snake | 44 gun Corvette | (elite crew) (24 pts) |
| (f) | Sea Lion | 44 gun Frigate | (elite crew) (24 pts) |

Voyage to the Bottom of the
Wind from the NW, blowing a fresh breeze.

This one is dedicated to Richard Basehart and David Hedison.

| | | | |
|---|---|---|---|
| (a) | Seaview | 120 gun 3 Decker SOL | (elite crew) (43 pts) |
| (a) | Flying Sub | 40 gun Frigate | (crack crew) (17 pts) |
| (b) | Mermaid | 136 gun 3 Decker SOL | (mutinous crew) (27 pts) |
| (s) | Giant Squid | 112 gun 3 Decker SOL | (green crew) (27 pts) |

Frigate Action:
Wind from the E, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (a) | Killdeer | 40 gun Frigate | (average crew) (15 pts) |
| (b) | Sandpiper | 40 gun Frigate | (average crew) (15 pts) |
| (s) | Curlew | 38 gun Frigate | (crack crew) (16 pts) |

The Battle of Midway:
Wind from the E, blowing a moderate breeze.

| | | | |
|---|---|---|---|
| (a) | Enterprise | 80 gun Ship of the Line | (crack crew) (31 pts) |
| (a) | Yorktown | 80 gun Ship of the Line | (average crew) (27 pts) |
| (a) | Hornet | 74 gun Ship of the Line | (average crew) (24 pts) |
| (j) | Akagi | 112 gun 3 Decker SOL | (green crew) (27 pts) |
| (j) | Kaga | 96 gun 3 Decker SOL | (green crew) (24 pts) |
| (j) | Soryu | 80 gun Ship of the Line | (green crew) (23 pts) |

Star Trek:
Wind from the S, blowing a fresh breeze.

| | | | |
|---|---|---|---|
| (a) | Enterprise | 450 gun Ship of the Line | (elite crew) (75 pts) |
| (a) | Yorktown | 450 gun Ship of the Line | (elite crew) (75 pts) |
| (a) | Reliant | 450 gun Ship of the Line | (elite crew) (75 pts) |
| (a) | Galileo | 450 gun Ship of the Line | (elite crew) (75 pts) |
| (k) | Kobayashi Maru | 450 gun Ship of the Line | (elite crew) (75 pts) |
| (k) | Klingon II | 450 gun Ship of the Line | (elite crew) (75 pts) |
| (o) | Red Orion | 450 gun Ship of the Line | (elite crew) (75 pts) |
| (o) | Blue Orion | 450 gun Ship of the Line | (elite crew) (75 pts) |

CONCLUSION
Sail has been a group effort.

Ken Arnold Code
curses library (pu!)

AUTHOR
Dave Riggle

CO-AUTHOR
Ed Wang

REFITTING
Craig Leres

CONSULTANTS
Chris Guthrie
Captain Happy
Horatio Nelson
Nancy Reagan
and many valiant others...

REFERENCES
Wooden Ships & Iron Men, by Avalon Hill
Captain Horatio Hornblower Novels, (13 of them) by C.S. Forester
Captain Richard Bolitho Novels, (12 of them) by Alexander Kent
The Complete Works of Captain Frederick Marryat, (about 20) especially
Mr. Midshipman Easy
Peter Simple
Jacob Faithful
Japhet in Search of a Father
Snarleyyow, or The Dog Fiend
Frank Mildmay, or The Naval Officer

SEE ALSO
midway(PUBLIC)

BUGS
Probably a few, and please report them to "riggle@ernie" and "edward@arpa."

NAME

snake, snscore - display chase game

SYNOPSIS

/usr/games/snake [ -wn ] [ -ln ]
/usr/games/snscore

DESCRIPTION

Snake is a display-based game which must be played on a CRT terminal from among those supported by vi(1). The object of the game is to make as much money as possible without getting eaten by the snake. The -l and -w options allow you to specify the length and width of the field. By default the entire screen (except for the last column) is used.

You are represented on the screen by an I. The snake is 6 squares long and is represented by S's. The money is $, and an exit is #. Your score is posted in the upper left hand corner.

You can move around using the same conventions as vi(1), the h, j, k, and l keys work, as do the arrow keys. Other possibilities include:

sefc These keys are like hjkl but form a directed pad around the d key.

HJKL These keys move you all the way in the indicated direction to the same row or column as the money. This does not let you jump away from the snake, but rather saves you from having to type a key repeatedly. The snake still gets all his turns.

SEFC Likewise for the upper case versions on the left.

ATPB These keys move you to the four edges of the screen. Their position on the keyboard is the mnemonic, e.g. P is at the far right of the keyboard.

x This lets you quit the game at any time.

p Points in a direction you might want to go.

w Space warp to get out of tight squeezes, at a price.

! Shell escape

^Z Suspend the snake game, on systems which support it. Otherwise an interactive shell is started up.

To earn money, move to the same square the money is on. A new $ will appear when you earn the current one. As you get

richer, the snake gets hungrier. To leave the game, move to the exit (#).

A record is kept of the personal best score of each player. Scores are only counted if you leave at the exit, getting eaten by the snake is worth nothing.

As in pinball, matching the last digit of your score to the number which appears after the game is worth a bonus.

To see who wastes time playing snake, run /usr/games/snscore .

FILES

/usr/games/lib/snakerawscores database of personal bests
/usr/games/lib/snake.log log of games played
/usr/games/busy program to determine if system too busy

BUGS

When playing on a small screen, it's hard to tell when you hit the edge of the screen.

The scoring function takes into account the size of the screen. A perfect function to do this equitably has not been devised.

NAME

trek - trekkie game

SYNOPSIS

/usr/games/trek [ [ -a ] file ]

DESCRIPTION

Trek is a game of space glory and war. Below is a summary of commands. For complete documentation, see Trek by Eric Allman.

If a filename is given, a log of the game is written onto that file. If the -a flag is given before the filename, that file is appended to, not truncated.

The game will ask you what length game you would like. Valid responses are "short", "medium", and "long". You may also type "restart", which restarts a previously saved game. You will then be prompted for the skill, to which you must respond "novice", "fair", "good", "expert", "commadore", or "impossible". You should normally start out with a novice and work up.

In general, throughout the game, if you forget what is appropriate the game will tell you what it expects if you just type in a question mark.

AUTHOR

Eric Allman

SEE ALSO

/usr/doc/trek

COMMAND SUMMARY

```
abandon                     capture
cloak up/down
computer request; ...       damages
destruct                    dock
help                  impulse course distance
lrscan                move course distance
phasers automatic amount
phasers manual amt1 course1 spread1 ...
torpedo course [yes] angle/no
ram course distance         rest time
shell                 shields up/down
srscan [yes/no]
status                terminate yes/no
undock                visual course
warp warp_factor
```

NAME

ttt - tic-tac-toe

SYNOPSIS

/usr/games/ttt

DESCRIPTION

Ttt is the X and O game popular in the first grade. This is a learning program that never makes the same mistake twice.

Although it learns, it learns slowly. It must lose nearly 80 games to completely know the game.

FILES

/usr/games/ttt.k learning file

NAME

warp - a real-time space war game

SYNOPSIS

warp [options]

DESCRIPTION

Warp is a real-time space war game that requires skill and quick thinking. "Real-time" in this context means that the enemies keep moving (and shooting) even if you don't. A unique feature of warp is that blast propagates; it is unhealthy to remain near things that are in the process of blowing up. If a given universe is above a critical density it may chain react. Scoring is like many popular arcade games--there are multiple waves which get harder and harder as you go along. Nobody has ever maxed out the scoreboard without cheating.

Unlike many space-war games, warp is not simply a shooting gallery. Along with phasers and photon torpedoes, you have tractor beams and a cloaking device. Skill in navigation is important. It helps to be schizophrenic, because you must manage an Enterprise and a Base simultaneously. And enemies do not simply shoot back. You can get tailed, absorbed, snuck up upon, hemmed in, rammed, loved to death, reprimanded for destroying civilized life, dragged around, robbed, damaged and eaten. And if you should happen to get bored by the enemies (a trifle unlikely), you can always watch the interesting star patterns. In fact, you'll have to, since your tactics will depend upon what kind of universe you find yourself in.

Warp is played in a double wraparound universe, i.e. the bottom is connected to the top, and the right is connected to the left. You need a crt with random cursor addressing and at least 24 lines by 80 columns. For more information about about how to play, simply run warp and say "y" when it asks if you want to see the instructions. There is also a single-page command summary that you can get while playing by typing a "?".

Command line options include:

-b Put warp into beginner mode. Makes the difficulty increase more slowly, but penalizes you for it.

-d<n>
Sets the initial difficulty to n.

-l Play a low-speed game. Changes the basic cycle time from 1 second to 2 seconds. This switch is automatically set at baud rates below 2400. You may want to

set it at higher speeds if your terminal cannot keep up with the output. (This should never happen on BSD systems, which have an IOCTL call to determine output queue length.) Because this makes the game easier, a separate scoreboard is kept for low-speed games.

-m Terminal has a meta key which turns on the eighth bit. Ordinarily the eighth bit is stripped in order to ignore parity. Metacharacters will appear to the keymap as prefixed with a ^A, and will subsequently have the same effect as a control character, unless otherwise mapped.

-s Just prints out the scoreboards and saved games and then exits.

-v Prints out the version number.

-x Play an experimental game. This causes warp to ignore any saved game, and disables the ability to save the current game. Thus you can play around with something or show warp to someone without jeopardizing a currently saved game.

ENVIRONMENT

WARPMACRO

If defined, names a file containing keyboard mappings and macros. If not defined, the value %X/Kbmap.%{TERM} is assumed. The macro file contains lines of the following form:

<keystroke-sequence> <whitespace> <canonical-keystroke-sequence>

You may use certain % interpolations and ^<letter> control characters. For possible % interpolations see warp.h. Sequences in the canonical-keystroke-sequence bounded by ^(...^) are subject to reinterpretation via the keymap. This file has two major uses. First, you can set up your commands to use any kind of prefix key your terminal might have, or change the key bindings in any other way you choose. Second, you can define arbitrary macros, such as this:

# define Corbamite maneuver = DDllllll

AUTHOR

Larry Wall <lwall@sdcrdcf.UUCP>

FILES

~/.fullname, if full names aren't in /etc/passwd

DIAGNOSTICS

Generally self-documenting, as they say.

BUGS

Addicting. At the end of a wave, all you have to do to keep going is hit a space. You see the message "Hit space to continue" and automatically hit space. About 2 seconds later you remember you wanted to go home, but by then it's too late to escape without penalty.

You can't kill a backgrounded warp process directly, because it is running setuid. You have to use the killer built in to warp.

Now that there is a space amoeba, there ought to be tribbles. But it might be too much trouble...

NAME

words - word game

SYNOPSIS

/usr/games/words

DESCRIPTION

Words prints all the uncapitalized words in the word list that can be made from letters in string.

FILES

/usr/dict/words the regular word list

BUGS

Hyphenated compounds are run together.

NAME

worm - Play the growing worm game

SYNOPSIS

/usr/games/worm [ size ]

DESCRIPTION

In worm, you are a little worm, your body is the "o"'s on the screen and your head is the "@". You move with the hjkl keys (as in the game snake). If you don't press any keys, you continue in the direction you last moved. The upper case HJKL keys move you as if you had pressed several (9 for HL and 5 for JK) of the corresponding lower case key (unless you run into a digit, then it stops).

On the screen you will see a digit, if your worm eats the digit is will grow longer, the actual amount longer depends on which digit it was that you ate. The object of the game is to see how long you can make the worm grow.

The game ends when the worm runs into either the sides of the screen, or itself. The current score (how much the worm has grown) is kept in the upper left corner of the screen.

The optional argument, if present, is the initial length of the worm.

BUGS

If the initial length of the worm is set to less than one or more than 75, various strange things happen.

NAME

worms - animate worms on a display terminal

SYNOPSIS

/usr/games/worms [ -field ] [ -length # ] [ -number # ] [ -trail ]

DESCRIPTION

Brian Horn (cithep!bdh) showed me a TOPS-20 program on the DEC-2136 machine called WORM, and suggested that I write a similar program that would run under Unix. I did, and no apologies.

-field makes a "field" for the worm(s) to eat; -trail causes each worm to leave a trail behind it. You can figure out the rest by yourself.

FILES

/etc/termcap

AUTHOR

Eric P. Scott

SEE ALSO

Snails, by Karl Heuer

BUGS

The lower-right-hand character position will not be updated properly on a terminal that wraps at the right margin.

Terminal initialization is not performed.

NAME

wump - the game of hunt-the-wumpus

SYNOPSIS

/usr/games/wump

DESCRIPTION

Wump plays the game of `Hunt the Wumpus.' A Wumpus is a creature that lives in a cave with several rooms connected by tunnels. You wander among the rooms, trying to shoot the Wumpus with an arrow, meanwhile avoiding being eaten by the Wumpus and falling into Bottomless Pits. There are also Super Bats which are likely to pick you up and drop you in some random room.

The program asks various questions which you answer one per line; it will give a more detailed description if you want.

This program is based on one described in People's Computer Company, 2, 2 (November 1973).

NAME

zork - the game of dungeon

SYNOPSIS

/usr/games/zork

DESCRIPTION

Dungeon is a computer fantasy simulation based on Adventure and on Dungeons & Dragons, originally written by Lebling, Blank, and Anderson of MIT. In it you explore a dungeon made up of various rooms, caves, rivers, and so on. The object of the game is to collect as much treasure as possible and stow it safely in the trophy case (and, of course, to stay alive.)

Figuring out the rules is part of the game, but if you are stuck, you should start off with "open mailbox", "take leaflet", and then "read leaflet". Additional useful commands that are not documented include:

quit (to end the game)

!cmd (the usual shell escape convention)

> (to save a game)

< (to restore a game)

FILES

/usr/games/lib/d*