Games # 1

Quick Reference

Disclaimer

None of us claim that these utility programs are good for anything. If **you** think they are, great, but that is up to you to decide. If any, or all, of these programs don't work, that is your problem, not ours. If you lose a million dollars, or anything else, because one or all of these programs stuffs up, you are out of pocket the million, not us. If you don't like this disclaimer: tough. We reserve the right to do the absolute minimum provided by law, up to and including nothing.

In no event will Applix be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect in the software or its documentation.

This disclaimer has been provided in plain English, in keeping with Applix's policy of providing comprehensive, readable information about its products. It is basically the same disclaimer all the fancy, expensive overseas software packets provide, but without legal beagles mangling the English. Special thanks to Dave Horsfall for bringing this disclaimer to my attention.

Conditions of Sale

By purchasing† this software you agree to:

Assign Power of Attorney to the Company.

Be cited as a reference for sales purposes.

Franchise your computer to the Company.

Use only Applix brand software. Your 1616/OS EPROMS have been adjusted to operate only with Applix proprietary software, and any attempt to use any other brand will destroy your entire database, your disk drive, and the room the computer occupies.

Accept the built-in update policy. The ***modem***¢ program dials up Applix and checks for any upgrades. These are automatically *purchased and your credit card debited. The program earns a small sales commision.*

Deny the program is faulty. Should you find a software bug, you will be considered a dissatisfied user, who is likely to destabilise the corporate image. Should you attempt to disclose your dissatisfaction with any product or service, the Company will make every effort to preserve its credibility by destroying yours. This includes, but is not limited to, sabotaging your database, your bank accounts, your share portfolio, your credit rating, and your police records. In the unlikely event that an effective Data Protection Act ever exists, these sanctions may be replaced by hiring a hit man.

Relinquish all common law rights regarding consumer protection. Your citizenship. Your right to any intellectual property created using these programs. Your first born male child.

The site licence authorises one user at one location only to use the software. If you can afford to have two computers, you can afford to buy another copy.

Any software fault is deemed to be the fault of the user. The Help command activates a modem request for the audit of your tax returns since 1978.

The company admits no liability for the quality of its product, or for its ability to perform any function whatsoever (except act as a frisbie.)

†Aren't you glad you didn't actually **have** to purchase this shareware?

Notes on the shareware were rewritten to correspond somewhat with the actual source code, complete with gratuitous errors to confuse you, by Eric Lindsay.

Comments about this manual or the software it describes should be sent to either:

Eric Lindsay
6 Hillcrest Avenue
Faulconbridge 2776
NSW Australia
(047) 512258

Applix Pty Limited
Lot 1, Kent Street
Yerrinbool, 2575
N.S.W. Australia
(048) 839 372

ISBN 0 947341 ?? ?

A Note to Authors

This is an attempt to provide quick reference guides to the wide variety of Shareware and Public Domain programs made available for the Applix 1616 in the first seventeen Shareware disks.

It would obviously not be possible without the enthusiastic support of all of you who wrote or converted programs. In writing this material, I've generally based my descriptions on the source code, and on what has often been a very rushed play with the programs themselves. This means that I'm likely to have a lot of stuff wrong, so I'm asking for your help again.

Would you each look through my description of the programs you provided, and let me know what I have wrong, and what corrections should be made?

If any of you would like your address shown, please let me know. If you are asking for a contribution for your software, could you please let me know the suggested amount, and to where it should be sent.

Some of you have provided extensive documentation files for your programs. If you would like these included in the manual, would you let me know so that I can convert and laser print them. Many thanks are due to Michael Johnson, Conal Walsh and Greyham Stoney for the extensive documentation of their many excellent programs.

The Quibbles heading is perhaps somewhat inconsistent, as I've included suggestions for upgrades and changes. In many cases, things mentioned as bugs should be considered opportunities for others to expand the work you have done, not as things that don't work. If you think I'm in error in your case, please don't hesitate to contact me so I can make changes.

Finally, it is my firm policy on this set of manuals, to provide complete updated versions (or pages if appropriate) to **everyone** who contributes shareware for the Applix 1616. This is part of my way of saying "thank you" for your support.

Due to the number of programs accumulated in the first 17 Applix shareware disks, I've decided to split the shareware manuals. This one, obviously, is the first games manual. The others are a utilities manual, and a light, sound and printer manual.

To make it easier for readers, each program is on a separate un-numbered page (or pages), so you can file them in any sequence you like (they normally are printed in alphabetical order of the program name).

Eric Lindsay, 6 Hillcrest Avenue, Faulconbridge NSW 2776.
BH (02) 2189651 Mon to Thurs. Weekends (047) 512258

advent - classic text adventure game - Mark Harvey, Andrew Morton - SW#2

advent [-red]

Description

-r Resume a game saved previously, prompts for name of saved game.

-e Echoes typed input lines to standard error, for capture in a file. Handy for working out what moves you shouldn't have made, or mapping.

-d Debug option.

Hints

Use two word commands to explore your surroundings. Collect treasures (or anything else you can find, just in case you need them). Fight off dwarves, etc. Try to figure out *which* two word commands work. Check contents of bottles before you drink them.

Words that might work (based on a quick source scan) include:

blast, break, brief, calm, drink, drop, eat, feed, fill, find, foo, food, inventory, kill, lock, log, nothing, off, on, open, pour, quit, read, rub, say, scare, suspend, take, throw, wake, walk, wave.

Quibbles

Save game appears to work, but actually creates a 0 length file.
Random crashes in one of the mazes.

Associated files

`advent.exec, advent.c, advent.h, advent1.dat, advent2.dat, advent3.dat, advent4.dat, advent5.dat, advent6.dat, catalog.doc, database.c, eadvent.c, english.c, environ.doc, history.doc, itverb.c, makefile, saveav.c, turn.c, verb.c`

See also

para, **wanderer**

Distribution

Applix 1616 Shareware Disk #2 /advent

Author

Original version lost in the mists of mainframe history, original language unknown. Micro versions worked on by Robert Ward, L C Calhoun, Greg Huntzinger, Bob Withers, etc. I even had a version in Basic on my OSI.

Conversion to Applix 1616 by Mark Harvey and Andrew Morton

amv - animal, mineral, vegetable guessing game - Dave Fowler - SW#4

amv.bas

Description

Run SSBASIC, then load game with `load "amv.bas"`, and type `run`.

Asks how to identify animal you have selected, adds details to its data file `amv.dat`, which must be in the same directory.

Associated files

`amv.dat`

See also

Michael Johnson shareware disk

Distribution

Applix 1616 Shareware Disk #4 /fowler

Author

Dave A Fowler, P.O. Box 269, Coffs Harbour Jetty, NSW 2450
Suggested donation $5.

Conversion to Applix 1616 by D A Fowler

ball - pong ball game - Andrew Driver - SW#3

ball

Description

The first "hit the bouncing ball" computer game, based on the original *pong* video game that made Nolan Bushnell's fortune. Single paddle, and you must have a joystick.

Sounds a buzzer when you miss the ball, so turn down the sound on your first attempt!

s starts the game

e ends the game

The game includes a wonderful *auto* mode that increases the speed as you continue. It slips into this demo mode after you lose your balls, so you can always act macho, twiddle your joystick at random, and *pretend* you are great!

Associated files

`ball.xrel, ball.s, ball.mac`

See also

pcrip, **space-potatoes.bas**, **warship**

Distribution

Applix 1616 Shareware Disk #3 /games

Author

Software, Andrew Driver
Hardware (it *needs* a joystick!), John Taylor

bkg - backgammon board game - Mark Harvey, Andrew Morton - SW#2

bkg

Brief Description

Two player board game, played with pair of dice. Each player has 15 men on the board. You play black, 1616 plays white. Object is to move black players from low numbered points to high numbered points, with no backwards moves allowed. First to remove all men wins.

On starting, **bkg** asks for Return to seed the random number generator.

Show your moves with

starting point/*point on first die* Space Bar *starting point* /*point on second die.*

You can move either one or two men, but moves can only be by the number of points shown on each die, not by a combination adding to the same number.

Associated files

`bkg.xrel, bkg.exec, bkg.c, bkg.h, bkg1.c, eval.c, move.c, movegen.c, screen.c, ssutils.c`

See also

stars, **yahtzee**

Distribution

Applix 1616 Shareware Disk #2 /backgammon

Author

Bill Mahoney

Conversion to Applix 1616 by Mark Harvey and Andrew Morton

brickwalls - manoeuvring game - Dave Fowler - SW#4

brickwalls.bas

Description

Computer draws two brick walls on screen. Directions are controlled by the **two** players. Force your opponent into a corner, or to run into you for a win. The only rule is not to place a brick on top of another brick.

Runs pretty fast on a 15 MHz Applix! (This means I couldn't control it.)

Has list of rules and key commands.

Left player		Right player	
w	up	**[**	up
s	right	**'**	right
a	left	**;**	left
z	down	/	down

See also

Distribution

Applix 1616 Shareware Disk #4 /fowler

Author

Dave A Fowler, P.O. Box 269, Coffs Harbour Jetty, NSW 2450
Suggested donation $5.

Conversion to Applix 1616 by D A Fowler

chess

Description

`Chess` plays against itself or against the user. You can optionally invoke it with one or two numeric parameters. One parameter sets the program search time in seconds. Two parameters set tournament time controls, where the first argument is the number of allowed moves, and the second is the number of minutes allowed for that number of moves.

When invoked, `chess` will display a graphic chess board. Use the cursor keys to move the highlighted square to the selected piece, then press Enter. Move the selected highlight from there using the cursor keys, and press Enter when you reach the desired destination.

Press the Esc key to get to the command mode. Type `help` for more details.

Other commands available include:

beep	Beep after every move.
bd	Update board positions.
book	Turn off opening book moves.
both	Play both sides of the game.
black	Computer takes black.
level	Set time controls such as 60 moves in 5 minutes, etc.
depth	Change search depth, from the default maximum of 29 (normally the timer stops the search).
easy	Toggles easy mode (thinking on opponents time - default is on).
edit	User defines board positions. # clears board, c toggles colour, . exits setup.
force	Allows user to enter moves for both sides.
get	Retrieve a game from disk.
help	Displays commands.
hint	Program gives you tips.
list	Writes moves and some statistics to `chess.1st`.
new	Start a new game.
post	Prints principle variation and score during search - 100 is equivalent to a 1 pawn advantage.

random	Randomise move selection slightly.
remove	Undo last two moves (one per player).
reverse	White at top of board.
quit	Exit the game.
switch	Switch places with opponent.
undo	Undo last move, by one player.
white	Computer takes white.

Examples

Quibbles

Occupies a **lot** of memory. Cut your **/rd** ram disk down to 32k or 24k before trying to start it. On the other hand, it requires **some** ram disk for its own use. Takes a while to load and initialise the 90k from floppy.
Use `setdate` to establish a correct date prior to starting it, as `chess` times its moves from the system clock.
Start it from the `chess.shell` file, not from the executable.
Pawn promotion is only to queen.

Associated files

29 files, mostly `chess.*` and `shapes`, plus subdirectories `/chess/doc` (10 files), and `/chess/shapes` (15 files). You need to leave the `gnu-chess.book` file in the same directory as `-chess.xrel`, as it is used for the opening moves. Source code can be moved elsewhere if required.

See also

laser

Distribution

Applix 1616 Shareware Disk #11 /chess

Author

Ported from a UNIX ASCII text only version written by Stuart Cracraft, John Stanback, Ken Thompson, Jim Aspres, and GNU contributors, etc, 31 December 1987. This is the GNU version.
Applix 1616 port, cursor movement and graphics by Andrew Morton.

robot `cprog1  cprog2`

Description

The idea is to write a strategy for your robot, in C, and then let it "play" against a similarly written program. Takes about 30 seconds to initialise everything usually. There are a couple of sample programs available.

Examples

```
robot croc dodge
robot robby savage
```

Quibbles

A bit fragile if your C program is wrong.

Associated files

Robot files include `croc` (agressive), `dodge` (poor strategy), `robby` (a test crobot), `savage` (very offensive), `eyes.c` (indefensible)
atan.c, atan2.c, crobot.c, cycle.c, fire.c, graphics.c, graphics.h, lock.c, main.c, makefile, move.c, rfun.c, robot.c, robot.h, scan.c, scores.c, test.c

See also

I can’t think of anything remotely like it.

Distribution

Applix 1616 Shareware Disk #16 /crobots

Author

Conversion to Applix 1616 by Colin McCormack.

greed - eat numbered points - Andrew Morton - SW#16

greed [-s]

Description

Fills the entire 640 by 200 display with numbers, each treated as from 1 to 9. The object is to erase as much of the screen as you can by moving round using the numeric keypad to select one of 8 directions. Your score is the total of all the numbers you eat.

The complications in the game are that you can not cross an eaten area. Also, the distance you move each time is determined by the number you first erase on that move, which makes careful planning essential.

You can also use the standard UNIX editor keys (hjkl yubn) to move.

Ctrl l to redraw the display.

q to quit.

? for help.

Examples

Quibbles

Derived from UNIX, uses curses routines.
Far too addictive!
Much more subtle than it appears.
Kathy's only comment is she doesn't believe Andrew's claimed 33000 score.
Oh yes, the **-s** option probably resets the score.

Associated files

`greed.c, greed.doc, greed.hs, greed.xrel, makefile, term.c`

See also

star

Distribution

Applix 1616 Shareware Disk #16 /greed

Author

Matthew T Day, for Unix.
Conversion to Applix 1616 by Andrew Morton.

hexagon - block the opponent - Michael Johnson - SW#5

hexagon

Description

Hexagonal grid with red and green players. The object is to reach the other side before the other player can get to the other side. You can block each other.

Use numeric keypad in cursor mode to move the cursor, use `5` key to establish a position.

Quibbles

Not yet fully debugged; executable sample only so far.

See also

Distribution

Applix 1616 Shareware Disk #5 /intro

Author

Suggested donation ?. Michael Johnson, 11/12 Kokoda St, Wagga Wagga, NSW 2650 (069)255255 (home), (069) 230388 (work).

humour - miscellaneous files

Description

`Digital-north.text` tell north by digital watch, SW#5
`Odds-ends.text` unlikely 68040 instructions, SW#5
`Real_programmers` don’t eat quiche, SW#6

Author

infocom - assist in Infocom adventures - Colin McCormack - SW#6

infocom `filename`

Description

A clever little program which will provide hints, tips and helpful descriptions for all Infocom text adventure games.

To use it, you must transfer a legitimately purchased Infocom game into Applix disk format. Sorry, the help program is not available for any other computer. For obvious reasons, actual Infocom games are not sold for the Applix. You can however get assistance in the transfer from the Users Group.

Quibbles

Associated files

`infocom.xrel`, `infocom.doc` (Sorry, no source available). Tested and worked on the following:

`ballyhoo.bat, cutthroats.dat, deadline.dat, enchanter.dat, hitchhikers.dat, leather.dat, lurking.dat, moonmist.dat, planetfall.dat, seastalker.dat, sorcerer.dat, sorcerer.doc, starcross.dat, starcross.doc, suspended.dat, wishbringer.dat, witness.dat, zork1.dat, zork2.dat, zork3.dat`

See also

advent, **para**, **vtrek**, **wanderer**

Distribution

Applix 1616 Shareware Disk # 6/

Author

Separation of data files by Infocom hackers group.
Conversion to Applix 1616 by Colin McCormack.

karma

Description

Use an `mrdrivers` file that allows you a large ram disk, and copy both `ssbasic` and `karma.bas` to the **/rd** for best operation.

Two players, brown and yellow, struggle to obtain territory by increasing their happiness level. Gain happiness by placing your cursor on your household. When a house is happy enough, it explodes into gossip, and converts surrounding territory to a higher happiness level.

The keys include 1 capture all, 2 four corners, 3 two pies, 4 2500 points, h help and q quit. Move using cursor keypad, and 5 key to select position.

Associated files

`karma.bas`

See also

laser

Distribution

Applix 1616 Shareware Disk #4 /ashby

Author

Modified from *Compute*, December 1987.
Peter Ashby, Victor Harbour High School, Main Road, Victor Harbour, S.A.

laser

Description

Strategy game, similar to chess, played on a 9 by 9 board, with rows marked 1 to 9, and columns marked *a* to *i*, using red and green players. There are 18 playing pieces, of which 9 are unique. Instructions are in 640 mode, play is in 320 mode. Use an `mrdriver` to establish a large ram disk, and transfer all files (including `ssbasic`) to **/rd** for best results.

Objects are affected by the direction and movement of light from each players laser, which can be rotated four ways. Mirrors can reflect and split the beam. The white side of an object is reflective, but many objects (including your own) are destroyed by the laser beam. You have a beam-splitter, that will produce two beams (and is often more dangerous to your own side).

The object of the game is to destroy or take the king, a small diamond shape.

A straight object can be rotated to be either horizontal or vertical.

A diagonal object can be at 45^0 or 315^0 to horizontal, and can not be destroyed by lasers.

Triangular solid objects can be destroyed.

The hypercube does random moves of your own or an opposing piece, and is not affected by lasers.

The block object can "take" another piece by moving to the square occupied by that piece, however it can be destroyed by laser.

Move using the numeric keypad, select pieces with the `5` key. Options are displayed with `0`.

R rotates selected item, 0, 5, 4 is sequence.
L laser fires, 0,L, 5 to select.
N new game.
Q quit.
L load a saved game.
S save present game.
dir will display directory.

Associated files

`laser.shell, laser.bas, get.xrel, put.xrel, shapes.xrel, ibmfont.xrel`

See also

karma, **chess**

Distribution

Applix 1616 Shareware Disk #4 /ashby

Author

Peter Ashby, Victor Harbour High School, Main Road, Victor Harbour, S.A.

life - graphics display, in C - Michael Johnson - SW#3, SW#5

life

Description

Prompts for screen size (smaller is faster), asks whether you want to use joystick or numeric keypad for the entry of the starting pattern (joystick is a lot easier).

You must draw a set of starting patterns on the display (otherwise it is very boring!) Use the joystick or numeric keypad. Keys move the cursor in 8 directions. Press **S** to leave pixels displayed. Press **c** (the default) to move without leaving pixels displayed.

When you have produced your starting pattern, start by pressing Enter.

Alt c stops the program. It prints the total number of generations, and then any key returns you to 1616/OS.

Space pauses the display.

The display wraps round. Uses the 320 by 200 display mode. Nice touch is having each generation change colour. Michael includes extensive notes about Conway's Game of Life, as discussed in *Scientific American*.

Examples

Quibbles

Slowness seems to often be a problem with **life**. Michael says the current algorithm is about 30 times faster than his original version. There is a very large character array set up at the beginning of the program, and the manipulation of this array is where the slowness inherently lies. You may speed it up a little, but the effort required would not justify the meagre improvement.

Associated files

```
life.exec, life.c, life.text
```

See also

kal, **life**

Distribution

Applix 1616 Shareware Disk #3 /games

Author

Conversion to Applix 1616 by Michael Johnson, 11/12 Kokoda St. Wagga Wagga NSW 2650 (069) 255255 (home,) (069) 230388 (work - a free call).
Suggested donation $10.

LIFE.XREL introduced by Michael Johnson

This is an implementation of Conway's *Game of Life*. The principles are that of birth, life and death. Each pixel on the screen represents a living organism. The basic principle is as follows:-

BIRTH

If an organism is non existent (pixel not lit) then it will be born if it has 3 living organisms adjacent to it.

LIFE

If an organism is alive and has 2 or 3 living organisms next to it then it remains alive.

DEATH

If an organism which is alive has more than 3 or less than 2 living organisms adjacent to it then it will die.

When the program is started it will ask for you the size of the screen you wish to use. The number you input will be clipped if necessary to keep it within the bounds which are given. You will then determine the method of input i. e. joy-stick or keyboard.

JOYSTICKS

The joysticks are scaled so that a full movement of the joystick will move the cursor from border to border. The program will read the joystick scale the value and then plot this point.

KEYBOARD

The keyboard input section uses the following commands:-

1 move cursor down and left 1 pixel.
2 move cursor down 1 pixel
3 move cursor down and right 1 pixel
4 move cursor left 1 pixel
6 move cursor right 1 pixel
7 move cursor up and left 1 pixel
8 move cursor up 1 pixel
9 move cursor up and right 1 pixel

S each pixel that the cursor passes over will be born
C each pixel that the cursor passes over will die

STARTING THE EXPERIENCE

When you have completed the input of the initial pattern you type <ENTER> to start the game. While the game is running you can pause it by pressing the space bar. The current generation will be completed and then the word PAUSE will be printed at the bottom of the screen. You may continue into the next and subsequent generations by pressing the space bar again.

TERMINATING THE EXPERIENCE

The game is terminated by the ALT-C key sequence. The program will complete the current generation, print FINISHED at the bottom of the screen and stop. Pressing any key will then drop you back into 1616o/s.

Michael Johnson

life - graphics display, in assembler - Gerhard Baumann - SW#14

lif

Description

You must draw a set of starting patterns on the display (otherwise it is very boring!) Use the joystick or numeric keypad. Keys move the cursor in 8 directions. Press Space to toggle and leave pixels displayed, and again to stop them being displayed. Uses the entire display.

When you have produced your starting pattern, start by pressing Enter.

h produces help.

q stops the program.

r refreshes the display.

s single steps to next generation.

Examples

Quibbles

Faster than Michael Johnson's version in C, but doesn't change colour in each generation. Start it using the shell file `lif.shell` for best results.

Associated files

`life.xrel, life.mac, table.mac, life.s, lif.shell`

See also

kal, **life**

Distribution

Applix 1616 Shareware Disk #14 /baumann

Author

Gerhard Baumann.

mastermind - number guessing game - Dave Fowler - SW#4

mastermind.bas

Description

The traditional number guessing game. Allows a range of positions from 4 to 10 numbers, and range of digits from 4 to 10.

Menu driven. A black indicates the right number in the right place, a white indicates the right number in the wrong place.

n New game
g Give up
s Stop playing

Suggested changes

Might be an idea to allow the use of letters, or even words, to expand the use of the game.

See also

Distribution

Applix 1616 Shareware Disk #4 /fowler

Author

Dave A Fowler, P.O. Box 269, Coffs Harbour Jetty, NSW 2450
Suggested donation $5.

Conversion to Applix 1616 by D A Fowler

names - alien name generator - Andrew Morton - SW#14

names [number]

Description

Generates number of pronounceable, but strange single word names, each separated by a new line. Maximum number is 30.

Examples

Quibbles

Associated files

names.xrel, names.c

See also

Distribution

Applix 1616 Shareware Disk #14 /

Author

Brett Slocum wrote this version, with changes suggested by Geoff Kimbrough.
Conversion to Applix 1616 by Andrew Morton.

nim - traditional math logic game - Dave Fowler - SW#4

nim.bas

Description

Traditional game of **nim**, playing against the computer.

Starting with the number 21, you and the computer each have to subtract a number between 1 and 3. Aim of the game is to avoid taking the last number.

I suggest you look up the algorithm, otherwise the computer will win every time.

Suggested changes

Allow different starting numbers.
Select who gets first turn.

See also

Distribution

Applix 1616 Shareware Disk #4 /fowler

Author

Dave A Fowler, P.O. Box 269, Coffs Harbour Jetty, NSW 2450
Suggested donation $5.

Conversion to Applix 1616 by D A Fowler

para

Description

Role playing adventure, in which everyone is out to get you. Ask for instructions, and lie whenever you can get away with it.

Para is short for paranoid, and that is how everyone acts.

Instructions? Why do you need them? I'm not going to tell you about that!

p displays statistics

Quibbles

They always kill me!

Associated files

`para.exec, para.c, paranoia.c` (paranoia.c is the original version), `paranoia.indx` (will build its own index file if this is not present), `paranoia.page` (there is now an .xrel version

See also

vtrek

Distribution

Applix 1616 Shareware Disk #2 /para

Author

Sam Shirley, *SpaceGamer/FantasyGamer* magazine, issue 77. C version for Vax under UNIX by Tim Lister. There was a CP/M version.

Conversion to Applix 1616 by Andrew McNamara

space-potatoes - they chase you - Dave Fowler - SW#4

space-potatoes.bas

Description

The space-potatoes are out to get you. Lure then into black holes to destroy them.

You can only use the hyperspace escape 4 times.

+ You
X Space potato
? Fast and deadly
***** Convenient black hole
@ Space warp

Set the Num Lock key, then use the 2,4,6 and 8 keys to move in the direction of the arrows.
0 Hyperspace escape.
s Stops play, as does Space Bar.

Suggested changes

Keyboard response is far too slow.

See also

Distribution

Applix 1616 Shareware Disk #4 /fowler

Author

Dave A Fowler, P.O. Box 269, Coffs Harbour Jetty, NSW 2450
Suggested donation $5.

Conversion to Applix 1616 by D A Fowler

star

Description

Star is a board game, played on a three by three numbered grid. Each position can contain a visible star, or an invisible star. When the game commences, the middle star (position 5) is visible, and all others are invisible. The object of the game is to reverse this position.

The only rule: You can only select a star which is visible.

a aborts the game.
Esc quits, and returns to operating system.

If you select a star using a number key, it turns invisible. If you select an invisible star, you get an error message.

When you select a star, the stars surrounding it change their state, as follows:

Select	Change	Select	Change	Select	Change
1	2,4,5	2	1,3	3	2,5,6
4	1,7	5	2,4,6,8	6	3,9
7	4,5,8	8	7,9	9	5,6,8

This makes selection of the correct moves logical, but frustrating. When you complete the game, it reports the number of moves required.

Curiosities

The colour scheme suited Michael's niece. Michael is now adding a user definable `.cfg` file to set the default colour scheme.

Associated files

```
star.exec
star.c
star.text
```

See also

bkg, **yahtzee**

Distribution

Applix 1616 Shareware Disk #1 /games

Author

Original version was called ‘Star Shoot’, and was written by Craig Barratt on the ancient 2650 CPU. Michael modified this system to use 2650 graphics circa 1977.

Applix 1616 version by Michael Johnson, 11/12 Kokoda St. Wagga Wagga NSW 2650 (069) 255255 (home,) (069) 230388 (work - a free call). Suggested donation $10.

INTRODUCTION by Michael Johnson

1616 version of a very old game that I used to play on a 2650 based computer system (1973). The game is ideal for learning how to think in a forward and logical manner. I enjoyed writing it and that's what counts.

THE BOARD LAYOUT

The board has 9 squares which are numbered from 1 to 9. Each square contains a star which is either visible or invisible. Star number 5 is visible when the game starts and all other stars are invisible.

THE OBJECT

The object of the game is to reverse this situation i.e. star 5 is invisible and all other stars are visible. The following diagrams will demonstrate these conditions.

a '-' means a star is invisible
a '*' means a star is visible

start position	star numbers	end position
- - -	1 2 3	* * *
- * -	4 5 6	* - *
- - -	7 8 9	* * *

THE MOVES

If you hit a star then that star turns invisible, it will also affect other stars in the following ways:-

star 1 — star 1 will be invisible
stars 2, 4 and 5 will change state

star 2 — star 2 will be invisible
stars 1 and 3 will change state

star 3 — star 3 will be invisible
stars 2, 5 and 6 will change state

star 4 — star 4 will be invisible
stars 1 and 7 will change state

star 5 — star 5 will be invisible
stars 2, 4, 6 and 8 will change state

star 6 — star 6 will be invisible
stars 3 and 9 will change state

star 7 — star 7 will be invisible
stars 4, 5 and 8 will change state

star 8 — star 8 will be invisible
stars 7 and 9 will change state

star 9 star 9 will be invisible
stars 5, 6 and 8 will change state

If you are hopelessly lost then press A to abort. This command will take you back to the start but will not reset the attempts counter.

If you wish to exit from the game (totally flustered!!??) then press the escape key. This will return you to 1616o/s.

THE RULES
You can only hit a star which is visible. (This is the only rule)

solitaire - jump the peg - Michael Johnson - SW#5

solitare

Description

Take a red peg, and move it into a blue hole, by jumping another red peg either vertically or horizontally. The peg you jump over is removed. Object is to remove all pegs but one, the problem is selecting the correct pattern of moves required to do so, without having pegs left alone and untouchable.

Esc exit key.

A abort this pattern, reset board, but move counter is not changed.

Move your cursor (white diamond) using the numeric keypad; you can move the cursor in any direction. Select a peg using the 5 key. Cursor changes to black. Deselect by press 5 again. Select your direction to move with the cursor keys, remembering you can only move horizontally or vertically.

This program provides nine types of solitare board, which are selected from a menu. The games available are the English, the French, the English Cross, the French Cross, the Corsair, the Octogon, the World, the Apostles, and the letter E. You can obtain instructions on all these games from the menu.

Suggested changes

Associated files

```
ccsolitare.shell, solitare.c, solitare.doc, solitare.h,
solitare.inst1, solitare.inst2, solitare.inst2, solita-
re.inst3, solitare.xrel, solitare1.c, solitare2.c
```

Distribution

Applix 1616 Shareware Disk #5 /solitare

Author

Michael Johnson, 11/12 Kokoda St, Wagga Wagga, NSW 2650, (069) 255255, Suggested donation $10.

syllogy
sylloword

Description

The BASIC version, `syllogy`, is a test run at a game to teach simple analogy and logic to children. It is a game that learns the meanings of sentences as you go along, giving a simple example of how expert systems or a.i. programs might grow.

The C version, `sylloword`, is more complex and alterable. The program starts with a menu, which will also appear if you enter a null (empty) string. From the menu you can exit, enter the actual program, load a database of known words, save a database, load a list of words to skip or ignore, or save a skip list.

The program already knows to ignore the words a, an, and, the, with, have. It already knows that the following are questions if at the start of a line (and ignores them when elsewhere in a line) do, does, has, will, can, is.

You can add words to the skip list by starting a line with `miss+`. You can see both the system and skip words by typing `list-`, and see the rest of the database words by typing `list+`. Ask for `help` if uncertain.

Start a few sentences with `what` or `who` to test it.

Suggested changes

Associated files

```
ccsylloword.shell, syllogy.bas, syllogy.doc, sylloword.c,
sylloword.doc, sylloword.h, sylloword.xrel, sylloword1.c
```

Distribution

Applix 1616 Shareware Disk #5 /sylloword

Author

Based on *Amstrad Omnibus* version,
Michael Johnson, 11/12 Kokoda Street, Wagga Wagga, NSW 2650, (069) 255255. Suggested donation $10.

tictactoe

Description

Simple text version of the game, originally intended to use curses.

Examples

Quibbles

Code is supposed to give `bus error (core dumped)` message if you beat it.

Associated files

`tictactoe.c, tictactoe.xrel`

See also

Distribution

Applix 1616 Shareware Disk #8

Author

Warren Toomey, 1988
Applix 1616 port by Andrew Morton.

vtrek

Description

Played on an 8 by 8 quadrant grid, each composed of an 8 by 8 sector grid. Asks if you want instructions, requests your name, your ship's name (default is Kirk and Enterprise), and the difficulty level required.

The aim is to destroy the Klingon ships with phaser and torpedoes, before you run out of energy and torpedoes. You can replenish supplies by docking at a Starbase.

Commands

l Long range scan - shows contents of one quadrant in every direction as a three digit number. Never move to a quadrant without scanning for Klingons.
Hundreds digit is number of Klingons.
Tens digit is number of star base.
Single digits are number of stars.

h Hyperspace - travel to another quadrant. Select direction using the keys around the [s], as in *Edit*. Select warp factor (number of quadrants to cross) from 0 to 8.

Impulse power travel (within a sector). Direction is selected by using the numeric keypad (you can travel in 8 directions).

s Short range scan. Exact direction and distance of Klingons. Shown on screen as:
K = Klingon
S = Starbase
* = Star
E = *Enterprise* (or initial letter of your ship's name).
(E) = *Enterprise* with shields up.

u Up with shields (toggles shields). If your shields are at 0, you will be destroyed as soon as you move to a quadrant containing Klingons. Klingon fire drains your energy.

f Fix damaged devices faster than normal, at the expense of extra energy.

p Phaser fire (automatic targeting, but you must use lots of energy).

t Torpedoes. Set course using keys round [s]. A hit destroys Klingon, however the course must be correct.

r Redraw the display.

Damage reports on upper right. Less than 100 indicates some damage. Below 0 means the device won't work.

Quibbles

Does not correctly update the score file.
Lots of noise on the sound port prior to the first instruction.
Entirely black and white. Might be a nice project for someone learning C to add colour to the reports.
Reprogramming the letters S, K and E with *ship* shaped characters would be very nice also.

Associated files

`hiscore.c, main.c, makefile.mak, plot.c, sub1.c, sub2.c, termio.c, torp.c, vtglob.c, vtrek.c, vtrek.h, vtrek.help, vtrek.sc`

See also

Distribution

Applix 1616 Shareware Disk #1 /games/vtrek

Author

Original version public domain via UNIX networks.

Conversion to Applix 1616 by Mark Harvey

wanderer - graphic adventure treasure quest - Andrew Morton - SW#8

wanderer

Description

Collect **all** the treasure, then go through the exit. Watch out for falling rocks, bombs, etc. Your time is limited.

Use the cursor keys to move.

? Help.

0 Quiet.

l Loud.

! Look at map.

s Save game.

r Restore game.

q Quit.

~ Jump to next level.

Examples

Quibbles

Start it from the `wanderer.shell`.

Associated files

`credits, curses.h, display.c, edit.c, fall.c, game.c, help.c, hiscore, icon.c, jump.c, m.c, makefile, manifest. newchars.exec, newchars.s, read.c, readme, save.c, scores.c, wanderer.shell, wand_head.h, _wanderer.xrel`, plus a directory `/wanderer/screens`.

See also

infocom, para, advent

Distribution

Applix 1616 Shareware Disk #8 /wanderer

Author

Steven Shipway, 1988
Applix 1616 port by Andrew Morton.

warship - depth bomb the submarine game - Andrew Driver - SW#3

warship

Description

You control the horizontal movement of the warship using your joystick, and drop bombs with the joystick fire button. The bigger (or was it, the deeper?) the submarine you destroy, the greater your points.

If you hit **b** while the high scores are displayed, you are given an opportunity to enter a password. If you get the password correct, you can then change many of the parameters of the game action.

Quibbles

Led to an unprovoked breaking of my joystick button. Maybe a keyboard only version is also required? (see `Warship2!`) And I wonder how much Kathy charges for joysticks?

Doesn't seem to have an exit key. If you reset out of it in 1616/OS Version 4, it continues to multitask in background, which makes for a strange display!

Associated files

```
warship.xrel, warship.s, warship.mac, war_inter.s,
war_subs.s, warship.doc
```

See also

ball, **pcrip**, **warship2**

Distribution

Applix 1616 Shareware Disk #3 /games

Author

Andrew Driver

warship2 - depth bomb the submarine game, in C - Andrew Driver - SW#16

warship2

Description

You control the horizontal movement of the warship using your joystick, and drop bombs with the joystick fire button. The bigger (or was it, the deeper?) the submarine you destroy, the greater your points. Depth charges now weave from side to side as they fall (nice touch).

The revised version, in C, lets you select keyboard insttead of joystick. Space starts the game, and drops bombs. Arrow keys move your ship.

If you hit **b** while the high scores are displayed, you are given an opportunity to enter a password. If you get the password correct, you can then change many of the parameters of the game action. New version gives extensive (mostly depressing) statistics after each game.

Quibbles

Even nicer version of an already fine game.

Doesn't seem to have an exit key, but gives a clean exit.

Associated files

```
warship.xrel, warship2.c, warship2.xrel, war_boat.c,
war_bomb.c, war_stuff.c, war_sub.c
```

See also

ball, **pcrip**

Distribution

Applix 1616 Shareware Disk #16 /warship2

Author

Andrew Driver

wisdom - cute quotes - Andrew Morton - SW#16

wisdom `wisdom.txt wisdom.idx`

Description

Use the `wis.shell` file to start it. Gives a pithy quote, selected from the `wisdom.txt` file, using the index file `wisdom.idx` to select the quote to print.

Quibbles

You have to use the `windex` program to update the index file, if you add quotes. No big deal, just thought I'd mention it.

Associated files

`windex.c, windex.xrel, wis.shell, wisdom.c, wisdom.idx, wisdom.txt, wisdom.xrel`

See also

names

Distribution

Applix 1616 Shareware Disk #16 /wisdom

Author

Conversion to 1616 by Andrew Morton (probably)

yahtzee - complicated dice chance game - Andrew Morton - SW#6

ya.shell

Description

Complicated dice game, in which (as usual) the object is to get the highest score. Roll dice three times per turn per player. More than one person can play.

Enter	Begin to play
?	On line help
Space	Toggles between human and computer
$	Shell escape window
Ctrl L	Redraw the display
b	Book of rules
j	Down cursor
k	Up cursor
q	Quit
v	Version display

Quibbles

Needs to run from writable disk, so it can store scores, etc.

Requires `assign` in Eprom or as MRD. Uses UNIX "curses" screen control package, so it must be started from the shell file.

No way am I going to suggest changes to a game whose rules I don't understand!

Associated files

`yahtzee.exec`, `yahtzee.doc`, `yahtzee.hs`, `environ` and a sub-directory with file, as `terminfo/A1616`

See also

Distribution

Applix 1616 Shareware Disk #6 /yahtzee

Author

Came from UNIX world, somewhere, which explains some of the strange control keys.

Conversion to Applix 1616 by Andrew K P Morton

Summary

advent [-red]

amv.bas

ball

bkg

brickwalls.bas

chess

robot `cprog1  cprog2`

greed [`-s`]

hexagon

infocom `filename`

karma

laser

life

lif

mastermind.bas

names [`number`]

nim.bas

para

space-potatoes.bas

star

solitare

syllogy
sylloword

tictactoe

vtrek

wanderer

warship

warship2

wisdom `wisdom.txt wisdom.idx`

ya.shell

Table of Contents