

STOS USER GROUP

NEWSLETTER 2

A QUESTION OF SPEED

For everyone who has been waiting with bated breath for the STOS Compiler to arrive, latest news from Mandarin says it should be ready for release in February 1989.

The price will be around £20.

LISTINGS

All the listings from this newsletter (plus whatever else I can cram on the disc) can be obtained by sending £2.00 [or £1.00 plus your own disc] to the club address.

Have YOU written any short programs or routines which demonstrate any aspect of using STOS? If so please send them in to me for publication.

It would be a great help if both routine and documentation were on disc in ASCII format in addition to a print out of both if possible.

SOUNDS GOOD!

Many members have asked about using sampled sound with STOS, so much so that Mandarin are currently developing their own sampler by the name of **MAESTRO** which is due for release in the spring of 1989.

The system will be in two parts: available either separately or bundled together.

The hardware is for those users without a sampler of their own and will cost £24.95.

The software which may be compatible with other hardware will also cost £24.95.

The bundled price is given as £49.90 (what... no discount for buying both together?... Pat!)

More details including proper specification in the next newsletter.

LETTERS

MARK PEARSON

A minor point - It would be nice to have the option of adding a title screen that does not 'spin' onto the screen.

{Can anybody help on this one? The offending routine seems to be nestling in a spaghetti of machine code..... Pat}

ROBERT BROWN

My interest is in two player games. I can't see in the manual any direct way of reading two joysticks - one in the mouse port and the other in port one.

{The joystick commands all apply to the normal (right hand) port. Can anyone supply a routine to simulate the commands for the mouse port?... Pat}

AARON FOTHERGILL

How can I directly access the ST's speaker so that I can replay samples (ie. direct clicking of the speaker at various amplitudes)?

How can I put blocks of graphics onto the screen without their going on in XOR mode (thus everything showing through wherever there's black used) without having to define two black colours or blocking out the area behind it? {Try using GR WRITING, page 133 of the manual... Pat}.

Why won't the graphics functions work properly in XOR mode (they just work black and white)? {As above - are you getting confused between the text writing modes on page 159 and the graphic writing modes on page 133?... Pat}

Why isn't there a decent text function in STOS (see GFA Basic for details)?

DAVID THOMSON

I would like to bring to the attention of the STOS programmers the inability of their program to operate correctly with certain

TOS patches installed. I carry out my development work on a Mega 4 ST with an SH205 hard disk. At boot time I load a wide range of OS patches such as a replacement file selector, GDOS, CYBERSND, MEGMATIC, L-CACHE etc. With GDOS loaded the drawing primitives in STOS would not work. This was corrected by the removal of GDOS but I require GDOS when using CAD-3D 2.0 and DEGAS Elite. I have recently discovered that another of my TSR's is causing problems with collision detection but as yet have not identified which one. Is there any possibility that such problems will be corrected. Please do not take these minor complaints too seriously. I am very impressed with STOS and look forward to the release of the compiler and 3D graphics extension with impatience.

KEVIN PARRY

The program is very exciting but I was disappointed to find procedures only catered for by GOSUB. I thought I had left those behind years ago. Why not have multi-line functions and procedures like most recent basics. Something that could be catered for by an *.EXA file?

PETER RICE

I have only just purchased STOS but my first impressions are of its power and ease of use. Apart from its enormous graphics facilities (the only machine with Basic installed to even scratch the surface of such power was the Memotech, much more powerful than MSX or Atari 8 bit) the language even combines the best features of Sinclair basic with those of Microsoft. I don't miss the procedures of the BBC. It has always been an overstatement to emphasise these.

A modular, structured approach to basic programming has always been just as easy without them. What would have been a nice touch would have been 'print at' rather than 'locate:print'. And with such power could STOS have included the record input/output structures for filing that PL/I uses. I am sure

LETTERS

It could. These are not complaints however. How could anyone complain with such a language, or should I say dialect, at ones fingertips.

{My Ideal would be for both line numbers and named subroutines. Both come in useful at different times.... Pat}

RICHARD TOOHEY

One bug/mistake I have discovered is the =HUNT function. The syntax is X=HUNT(start TO end,a\$) and STOS will search from start to end for a\$. Start must be greater than 15. This, I presume, is because STOS assumes references between 0 and 15 to mean the banks. So X=HUNT(0 to 65535,"Hello") will produce illegal function call or Memory bank not reserved (if start is between 1 and 15).

MARK BLENKINSOP

I have a Radio Shack TRS80 colour graphic printer connected to my Atari 1040 ST through a parallel interface. The printer also has a serial interface, RS-232-C using data and busy, 600 baud, 7 bit character, no parity, two stop bits. Can I get a hardcopy of say a particular sprite on this printer?

LEE GROVES

I would like to warn STOS users of a possible dangerous bug within the STOS prog. I have found that if you return to the desktop by using the SYSTEM command while a floppy disc is in the drive and NOT WRITE PROTECTED the data on that disc will possibly be ruined beyond recovery. This has happened a couple of times but may not do it every time. I haven't been able yet to duplicate the conditions which cause this but if the following points are followed the bug should not cause any problem.

1. Frequently make a back up copy of the program etc. you are working on.
2. Keep the floppy disk(s) which are left in the drive(s) write-protected when returning to

the desk top via the SYSTEM command.

RALPH VINCENT FLIGHT SYSTEMS, WEST EMERIL AVE. SALT LAKE CITY, UTAH, USA.

My background in programming has been with many Basics and Modula-2. For the last two years or so I have been using GFA Basic and have found the translation to STOS fairly easy although it is a little cumbersome to use line numbers again.... (HA!)

I am in the process of writing a commercial fantasy role playing game using the Tunnels & Trolls game system. I have switched to STOS because of the graphics and such and hope to 'Jazz' up the game. So any advice you can give me on using MAP.ACB and using it in a STOS program would speed up my production tremendously! I have not attempted to use sprites and other frills so far because of lack of experience, but with STOS it seems it will be very easy.

Do you plan on coming out with a compiler for STOS? I also noticed the memory bank 4 is reserved for some type of 3D extension. Is that also in the works?

I would like to correspond with other STOS users and they are invited to write to me for the exchange of information and possibly a joint commercial project or who knows what!

ALAN CARPENTER

I managed finally to get a runnable program to work but doesn't it take a long time to load it all! I can't wait for the compiler to come out - any news about when it will appear?

I'm busy writing a game at present and found a tip that may be of use to others. The problem was that having generated a Degas picture I wanted to put sprites in

LETTERS

various positions on the screen. Trouble was that it had to be word (16 pixels) aligned and I wanted, well, say, 12 pixels along. Well the answer is to use the hot spot in the sprite editor. If you find that you still can't get it right using the preset icons on the left hand side, then move the mouse to the position that you want and click the mouse button and the hot spot will move to ANY position within the sprite edit field.

The point to remember is that the x,y co-ordinates that you use for the sprite position is the point where the hot spot pixel within your sprite will be positioned.

STEPHEN HILL

Whilst I don't have a copy of ST REPLAY here are a few suggestions as to the possible causes of Andrew James' problems.

1. The error might have been caused by a difference in syntax between ST Basic and STOS Basic. This could conceivably lead to a bus error when the program is run as the STOS syntax checker still has a few rough edges. If possible, try rewriting the program slightly to see what happens.

2. The program may have attempted to call some machine code which used a TRAP instruction. Traps with numbers 3, 5 & 7 are used by STOS Basic. (I'm told that Traps 4 & 6 got ripped out of the system by JAWX when we were not looking!) It's also possible that the machine code has been placed somewhere inappropriate and has been overwritten before it could be called. This could easily generate the problem Andrew describes.

3. The program may be calling a machine code routine using interrupts and these clash with STOS Basic. If STOS Basic still works after the call, this is fairly unlikely.

I read in the review of ST REPLAY (June 87 ST USER) that the sound it generated could be played independently of the cartridge. If

you can't find a music expert, get Andrew to send me a disc containing an example of the problem and I'll see what I can do.

ALAN NORTHCOTT

Thank you for the help with "SORT". I have now determined what I was doing wrong, although it may well be worth pointing out the results to others as the listing in the manual could give unexpected results.

The problems hinge around the fact that SORT sorts the whole array as dimensioned. As dimensioning a numeric array sets the elements to zero, any elements not subsequently assigned are zero and come first in the sorted array. This incidentally means the example in the manual does not work properly if the number entry is terminated early with a zero.

As I was trying to sort string arrays the apparent problem was more devious. Printing the first few elements after sorting an unfilled array results in blank lines (carriage returns only) as the unassigned elements are "null strings" which sort to the beginning of the array.

The answer of course, in either case is to dimension to the required size or if this is not possible, as in my case, check through the elements of the array until data is found (or otherwise determine the number of valid items) and move these to the head of the array if required with a simple loop. Obviously where zeroes can be valid entries in numeric arrays care must be taken not to lose them.

On another topic - although menus, mouse, windows etc all work well, STOS does not particularly assist in multiple choice eg. YES/NO/CANCEL boxes as in other Basics. As STOS is expandable can this enhancement be added. For general assistance in programming this sort of selection device, can I recommend you break down or adapt the ZOLTAR coding (lines 11005-11075) in the

SPIRAL

By P. J. T. BUTLER

I have enclosed this listing for a program I have written using STOS. I must agree with all the praise as to the quality of STOS. My only minor problem is that when constructing a REPEAT UNTIL loop you cannot have more than one condition.

ie.

10 repeat

20 ---

30 ---

40 until ANSWER=1 or ANSWER=2

This meant writing the program slightly different to the way I had planned by including two loops, one nested inside the other.

ie.

10 repeat

20 repeat

30 ---

40 ---

50 until ANSWER=1

60 until ANSWER=2

The other thing I would like to have seen is the inclusion of PROCEDURES as the course I am taking at the moment uses these quite extensively.

I hope you find the program I have written of interest. It is a different approach to the demonstration of the 'shift' command. The program uses menus to display the various options. I have included:

- ➔ 10 different palettes of colours
- ➔ 5 different shift rates
- ➔ Colour distribution can be calculated or entered manually.
- ➔ Spirals can be drawn at any angle with step sizes from 1-4.

SPIRAL

```

10 rem Spiral program with menus
20 rem *****
30 rem main program loop
40 rem *****
50 mode 0 : curs off : flash off : key off
60 gosub 11100 : rem set up initial palette
69 rem *****
70 rem These variables are required in case
none entered in the menu.
71 rem *****
80 COLYES=1 : STP=2 : QUIT=0
90 ANGLE#=119 : Z#=2
100 gosub 9000 : rem ** set up menus **
120 menu on 13
129 rem ***** Main loop- this
checks to see which menu bar the pointer is
clicked on and directs the program to the
appropriate subroutine.
130 repeat : rem ** main loop start
140 SUB=mnbar
150 on SUB gosub 10000,11000,13000,
13500,14000
160 until QUIT=1 : rem ** loop finish
170 menu off
180 default
190 end
8999 rem *****
9000 rem menu$ set up subroutine
9001 rem *****
9010 menu$ (1)=" Angle ",0,2
9020 menu$ (2)=" Palette ",0,4
9030 menu$ (3)=" Shift speed ",0,6
9040 menu$ (4)=" Col FX ",0,8
9050 menu$ (5)=" * ",0,10
9099 rem *****
9100 rem Angle menu$
9101 rem *****
9110 menu$ (1,1)="Angle",0,2
9120 menu$ (1,2)="Step ",0,2
9130 menu$ (1,3)="Draw ",0,2
9140 menu$ (1,4)="Quit ",0,2
9199 rem *****
9200 rem Palette menu$
9201 rem *****
9210 menu$ (2,1)=" One ",1,0
9215 menu$ (2,2)=" Two",2,0

```

```

9220 menu$ (2,3)=" Three ",3,0
9225 menu$ (2,4)=" Four",4,0
9230 menu$ (2,5)=" Five",5,0
9235 menu$ (2,6)=" Six",6,0
9240 menu$ (2,7)=" Seven",7,0
9245 menu$ (2,8)=" Eight",8,0
9250 menu$ (2,9)=" Nine",9,0
9255 menu$ (2,10)=" Ten",10,0
9260 for F=11 to 15
9270 menu$ (2,F)=" ",F,0
9280 menu$ (2,F) off
9290 next F
9299 rem *****
9300 rem Shift speed menu$
9301 rem *****
9310 menu$ (3,1)="1",0,6
9320 menu$ (3,2)="2",0,8
9330 menu$ (3,3)="3",0,6
9340 menu$ (3,4)="4",0,6
9350 menu$ (3,5)="5",0,6
9360 menu$ (3,6)=" Off ",0,6
9399 rem *****
9400 rem colyes menu$
9401 rem *****
9410 menu$ (4,1)="calc on",0,8
9420 menu$ (4,2)=" off ",0,8
9499 rem *** ? menu ***
9500 menu$ (5,1)=" ? ",0,10
9600 return
9999 rem *****
10000 rem angle step draw quit sub-this
subroutine finds which element the pointer is
on and directs the program to the next level
of subroutines,it also checks to see if 'quit'
has been used, sends back QUIT variable
i.e.(flag).
10001 rem *****
10005 SEL=mnselect
10010 on SEL gosub 10100,10200,15000, 10400
10020 if SEL=4 and left$(QUIT$,1)="Y" or
left$(QUIT$,1)="y" then let QUIT=1
10030 return
10099 rem *****
10100 rem enter angle sub - sends back
ANGLE# variable
10101 rem *****

```

SPIRAL

```

1010 pen 1
10120 windopen 1,7,10,25,4,13,4
10130 pen 7
10140 Input " Enter an angle please
";ANGLE#
10150 windel 1
10160 return
10199 rem *****
10200 rem enter step sub-sends back STP &
Z# variables
10201 rem *****
10210 pen 1
10220 windopen 1,3,10,35,4,13,4
10230 pen 7
10240 repeat
10250 repeat
10260 Input " Enter a step size please
(1to4)";STP
10270 until STP<5
10280 until STP>0
10290 windel 1
10300 If STP=1 then Z#=4
10310 If STP=2 then Z#=2
10320 If STP=3 then Z#=1.4
10330 If STP=4 then Z#=1
10340 return
10399 rem *****
10400 rem quit sub - sends back QUIT$
variable
10401 rem *****
10410 pen 1
10420 windopen 1,4,10,33,4,13,4
10430 pen 7
10440 Input " Are you sure you want to quit
(Y/N)";QUIT$
10450 windel 1
10460 return
10999 rem *****
11000 rem Palette sub - finds which line
pointer is clicked on and goes to subroutine
with appropriate palette selection. (more
palettes can be added with little pro-
blem,you must remember to
add any lines to the menu and also to the
line-on SEL gosub).
11001 rem *****
11010 SEL=mnselect

11020 on SEL gosub 11100,11200,11300, 11400,
11500, 11600,11700 ,11800, 11900,12000
11030 return
11099 rem *****
11100 rem Palette 1 sub
11101 rem *****
11110 palette $0,$702,$700,$730,$750,
$770,$470,$70,$75,$77,$57,$27,$7,$507,
$707,$704
11120 return
11199 rem *****
11200 rem Palette 2 sub
11201 rem *****
11210 palette $0,$100,$200,$300,$400,
$500,$600,$700,$111,$222,$333,$444,$555,
$666,$777,$767
11220 return
11299 rem *****
11300 rem Palette 3 sub
11301 rem *****
11310 palette $0,$31,$41,$51,$61,$71,
$170,$270,$ 370,$470,$570,$670,$770,
$660,$550,$440
11320 return
11399 rem *****
11400 rem Palette 4 sub
11401 rem *****
11410 palette $0,$7,$107,$207,$307,$407,
$507,$607,$707,$704,$604,$504,$514,$524,
$534,$544
11420 return
11499 rem *****
11500 rem Palette 5 sub
11501 rem *****
11510 palette $0,$7,$227,$447,$667,$73,
$373,$573,$773,$700,$720,$740,$760,$305,
$416,$536
11520 return
11599 rem *****
11600 rem Palette 6 sub
11601 rem *****
11610 palette $0,$705,$715,$725,$735,
$745,$755,$765,$775,$675,$575,$475,$375,
$275,$175,$75
11620 return
11699 rem *****
11700 rem Palette 7 sub

```

SPIRAL

```

11701 rem *****
11710 palette $0,$145,$245,$345,$445,
$925,$215,$105,$106,$8,$7,$104,$214,
$334,$234,1345
11720 return
11799 rem *****
11800 rem Palette 8 sub
11801 rem *****
11810 palette $0,$511,$531,$551,$351,
$41,$240,$440,$420,$300,$213,$123,$342,
$552,$642,$612
11820 return
11899 rem *****
11900 rem Palette 9 sub
11901 rem *****
11910 palette $0,$551,$550,$552,$660,
$661,$662,$663,$673,$763,$764,$770,$771,
$772,$773,$774
11920 return
11999 rem *****
12000 rem Palette 10 sub
12001 rem *****
12010 palette $0,$700,$710,$720,$730,
$600,$610,$620,$630,$500,$510,$520,$530,
$400,$410,$420
12020 return
12999 rem *****
13000 rem Shift speed sub-finds which line
pointer was clicked on and sets up shift
rate.SHF variable sent back
13001 rem *****
13010 SEL=mnselect
13020 If SEL<6 then shift SEL else shift off
13030 return
13499 rem *****
13500 rem Col FX sub-finds which line pointer
clicked on and sendsCOLYES variable back
this is a flag.A subroutine is usedif the no.of
colours is not to be calculated by the
drawsubroutine. n.b.this routine is used
before a drawing ismade therefore if no.of
colours is changed the patternwill have to
be redrawn.
13501 rem *****
13510 SEL=mnselect
13520 If SEL=1 then COLYES=1
13530 If SEL=2 then COLYES=0 : gosub 13700
13540 return
13699 rem *****
13700 rem colx# request sub - this asks the
user to Input the amountof colours they
require COLX# variable sent back.
13701 rem *****
13710 pen 5
13720 windopen 1,4,10,34,4,13,4
13730 pen 7
13740 repeat
13750 repeat
13760 Input " Enter amount of colours
please (3 to 15) ";COLX#
13770 until COLX#>2
13780 until COLX#<16
13790 windel 1
13800 return
13999 rem *****
14000 rem program by sub-displays program
details in window
14001 rem *****
14010 pen 5
14020 windopen 1,5,5,31,12,12,4
14030 pen 6
14040 print : print "Program written in STOS
basic"
14050 print " by P.J.T.Butler"
14060 print " from an original program by"
14070 print " B.I.D.Mellors"
14080 print " completed on 25-11-1988"
14090 pen 9
14100 print : print " Press mouse key to exit"
14110 repeat : until mouse key
14120 windel 1
14130 return
14999 rem *****
15000 rem draw sub-this is the heart of the
program,calculations anddrawing are all
done here.variables required are-COL-
YES,ANGLE#,Z#,STPvariables used are-
COLX#,COL#,X#,Y#,ANG#,A#(loop),P#,Q
#
15001 rem *****
15010 menu freeze : hide
15020 cls
15030 If COLYES then let COLX#=(360/
ANGLE#)

```

SPIRAL

```

15040 If COLX#>15 then let COLX#=COLX#/2 : goto 15040
15050 If COLX#<3 then let COLX#=COLX#*2 : goto 15050
15060 let COL#=1
15070 let X#=160 : let Y#=105
15080 let ANG#=ANGLE#*(pi/180)
15090 for A#=0 to (1200/Z#)/pi step ANG#
15100 let P#=(A#/STP)*cos(A#)+160
15110 let Q#=(A#/STP)*sin(A#)+105
15120 let COL#=COL#+1
15130 If COL#>COLX# then let COL#=1
15140 Ink COL#
15150 draw X#,Y# to P#,Q#
15160 let X#=P# : let Y#=Q#
15170 next A#
15180 show on
15190 menu on
15200 return
16000 rem *****
16010 rem ** THE END **
16020 rem *****

```

LETTERS

[Continued from page 4]

newsletter to give a general routine which can be MERGED into programs?

The particular trick which I, as an old 8-bit programmer find novel is the use of screen zones with SET ZONE and Z=ZONE(0) to automatically detect the presence of the mouse sprite and allow the options to be highlighted and UNTIL MOUSE KEY which is zero i.e. FALSE until a key is pressed is also an economy of coding to select the item.

I hope these points give you some assistance in producing the newsletter. Although time as always is in short supply I will try to contribute on points of interest from time to time.

LISTINGS

FOUR WAY

SCROLLING

Richard Vanner saved my sanity with this routine.

It allows you to have four full screens in memory and move smoothly across all of them without the need to 'flip' from one to another.

Now.... who can come up with a method of viewing just a section (ie. through a frame)? I keep coming a cropper when the viewing window is over a picture border.

Help!

```

100 fade 1 : wait 7 : key off : mode 0 : flash
off
120 reserve as screen 5 : reserve as screen 6
: reserve as screen 7 : reserve as screen 8
130 load "screen1.neo",5
140 load "screen2.neo",6
150 load "screen3.neo",7
160 load "screen4.neo",8
165 fade 1 to 5
170 XD=16 : YD=1 : X=0 : Y=0 : X2=320 : Y2=200
180 logic=back
190 if jleft=true and X>0 then X=X-16
200 if jright=true and X<320 then X=X+16
210 if jup=true and Y>0 then Y=Y-10
220 if jdown=true and Y<200 then Y=Y+10
230 gosub 240 : screen swap : wait vbl :
goto 190
240 screen copy 5,X,Y,X2,Y2 to logic,0,0
250 if X>0 then screen copy 6,0,Y,X,Y2 to
logic,X2-X,0
260 if Y>0 then screen copy 7,X,0,X2,Y to
logic,0,Y2-Y
270 if X>0 and Y>0 then screen copy 8,0,0,X,Y
to logic,X2-X,Y2-Y

```

ROLE PLAYING

Generating character attributes is simple with this short routine. Please feel free to adapt and use it in your own program.

```

10 rem CHARACTER GENERATOR
20 cls
30 rem ARRAY TO HOLD ATTRIBUTES
40 dim CH(6,10) rem 6 characters * 8 attr
50 rem
60 print " ST IQ LK CN DX CR SP CLASS
PROF"
70 for X=1 to 6
80 rem ROLL DICE
90 for N=1 to 8 : CH(X,N)=rnd(15)+3 :
CH(X,8)=100*CH(1,1) : next N
130 CH(X,9)=0
140 for N=1 to 7
150 if CH(X,N)<10 then print " ";
160 print CH(X,N);
170 next N
180 if CH(X,1)>11 and CH(X,2)>11 and
CH(X,3)>11 and CH(X,4)>11 and CH(X,5)>11 and
CH(X,6)>11 and CH(X,7)>11 then CH(X,9)=1
190 if (CH(X,1)>CH(X,2) or CH(X,1)>CH(X,3))
and CH(X,2)<10 and CH(X,9)=0 then CH(X,9)=2
200 if (CH(X,3)>CH(X,1) or CH(X,3)>CH(X,2))
and CH(X,9)=0 then CH(X,9)=4
210 if (CH(X,2)>CH(X,3) or CH(X,2)>CH(X,1))
and CH(X,9)=0 then CH(X,9)=3
220 if CH(X,9)=0 then CH(X,9)=2
230 CH(X,10)=rnd(4)
231 if CH(X,9)=1 then print " war/wlz";
232 if CH(X,9)=2 then print " warrior";
233 if CH(X,9)=3 then print " rogue ";
234 if CH(X,9)=4 then print " wizard ";
250 CH(X,10)=rnd(4)
260 if CH(X,10)=0 then print " thief "
270 if CH(X,10)=1 then print " hunter"
280 if CH(X,10)=2 then print " miner "
290 if CH(X,10)=3 then print " smith "
300 if CH(X,10)=4 then print " farmer"
310 next X

```

GRABBER

Graphics can take up a tremendous amount of memory so it is often better to store them on disc rather than in RAM.

This program offers a method of grabbing a section of a graphic and storing it in a random access file.

The string holding the screen section can then be called back at any time and printed to the screen or placed in a temporary buffer ready for instant use.

I have deliberately left the routine as general as possible so that it may be adapted.

```

10 rem store partial screens as strings
11 hide : flash off : mode 0 : cls : key off
20 rem
140 window 1,1,20,39,4
141 window 1
150 input "R(ead) or W(rite) or
G(rab)";CHOICE$
160 if CHOICE$="W" or CHOICE$="w" then
gosub 1000
170 if CHOICE$="R" or CHOICE$="r" then
gosub 2000
175 if CHOICE$="G" or CHOICE$="g" then
gosub 1000 : gosub 3000
180 if CHOICE$<>"W" and CHOICE$<>"w"
and CHOICE$<>"R" and CHOICE$<>"r"
and CHOICE$<>"G" and CHOICE$<>"g" then
goto 150
190 goto 150
999 rem *****
1000 rem load a screen
1010 print "put main plc store in A then
anykey" : wait key : clw
1020 F$=file select$("*.*")
1030 load F$
1035 gosub 3000
1040 PC$=screen$(physic,X1,Y1 to X2,Y2)
1050 print "put file disc in A then anykey" :
wait key
1060 input "Small pic number";PCNO
1070 rem open random access file
1080 open #1,"R","smallpic"
1090 rem assign field
1100 field #1,20000 as PC$
1110 rem write the string to disc
1120 put #1,PCNO
1130 close #1
1140 return
1999 rem *****
2000 rem read from the file
2010 cls : window 1,1,20,39,4 : window 1 :
print "put file disc in a" : wait key
2020 rem open file
2030 open #1,"R","smallpic"
2040 rem assign field
2050 field #1,20000 as PC$
2060 pof(#1)=1
2070 rem get picture number
2080 input "which picture";PCNO
2090 rem retrieve picture from random file
and place in string pc$
2100 get #1,PCNO
2110 rem close file before I forget
2120 close #1
2130 rem print string to screen
2140 screen$(physic,0,0)=PC$
2150 return
2999 rem *****
***** rem
*****
3000 rem grab a portion of the screen as
defined by user
3025 X1=0 : Y1=0 : X2=64 : Y2=64 : x mouse=0
: y mouse=0
3040 rem move box with mouse
3050 repeat
3055 X1=x mouse : Y1=y mouse : X2=X1+64 :
Y2=Y1+64
3056 gr writing 3
3060 if X2<320 and Y2<200 then box X1,Y1 to
X2,Y2
3080 if X2<320 and Y2<200 then box X1,Y1 to
X2,Y2
3090 until mouse key=1
3100 box X1,Y1 to X2,Y2
3300 return

```

GOODIES AND THINGS

DISCS

- 1 to 5 discs..... £1.25 each
- 6 to 10 discs..... £1.20 each
- 11+ discs..... £1.15 each

These are good quality unbranded D/sided discs with labels.

LABELS

Self adhesive, assorted colours, 3.5" disc size.

- 10 for 50p
- 50 for £1.75
- 100 for £3.25

PUBLIC DOMAIN SOFTWARE

PD1

Virus killers, disc accessories, ramdiscs including UNDELETE, and a prog to get hard copy of disc directories.

PD2

Desk accessories, 20 odd assorted items to make life easier and cause a few smiles.

PD3

Miscellaneous utilities including documentation to make full use of ARC.TTP and lots of other useful progs.

PD4

Assorted games eg. cards, checkers, chess etc.

PD5

Some Fast Basic games and run file also INVADERS with mono emulator, RIPCORN and MEGAROID.

PD6

STAC (ST Adventure Creator) demo

These PD collections cost £2.00 each or £1.00 if you supply your own disc. They are supplied in single sided format on d/sided discs.

Contributions of useful/fun PD always welcome to build up the selection.

SHAREWARE

These games were written using STAC and are being distributed on behalf of the authors. Each copy of the game sold earns £1.00 or more for the author.

Contributions welcome for this section. Games/Utilities welcome whatever they are written in eg. STOS, other Basics, Machine Code, C etc.

SH1 - THE GRIMOIRE by R W Lee

As a student at Poly you need to find a Grimoire which is a book of ancient evil magic to aid you in your studies. Text/graphic/aural adventure.

SH2 - THE TRIALS OF CHRONOS by L R Faux/S & D Gray

As an archeologist on a field trip to Egypt you stumble across an ancient map of the Valley Of Kings. The map details an unmarked tomb. Text/graphics adventure.

These shareware discs are £3.00 each (£1.00 of which goes to the author) or £2.00 if you supply your own disc. They are NOT public domain so please make sure the author receives a cut if you want to pass them around. Help stamp out the piracy plague by beginning at home with a conscience!

ORDERING

To the best of my knowledge, all the programs on the PD discs are true Public Domain. If anyone knows otherwise please let me know at once so that the offending item can be deleted from the list. All prices include postage and packing. Cheques, POs or stamps for small amounts to: PAT WINSTANLEY,