

STOS USER GROUP

Pat Winstanley

first edition

EDITORIAL

Welcome to the first issue of the STOS User Group newsletter, a pretty rushed affair this time in an attempt to get support for STOS going as soon as possible.

Some of you have never used Basic before - don't worry - we won't go over your heads. Those of you who eat programming for breakfast may find yourselves pushed to new limits by the questions asked... "Why can't you...?!"

Everyone is vital to the group, beginner and expert alike, so please join in as much as possible. Any contributions are welcome, the size of the newsletter being to some extent flexible depending on available material.

An important function of the group is to put members with different skills in touch with each other, so if you're a novice programmer with a great eye for graphics or a hardened hacker who's tone deaf, make your skills and needs known through these pages. Most successful programs are team efforts these days so here's your chance.

I'd appreciate articles, listing etc on disc (ASCII or PROTEXT if possible) backed up by a printout. If you can't do so, don't worry, I still want your contributions to make the STOS newsletter the most successful of its kind.

Best wishes

HOW THE STOS GROUP WORKS

1. Everybody joins for the full year (September to August), those joining from 1st June being allowed £1 off the annual fee in recognition of the shorter period of helpline use.
2. Members joining during the year receive any current and back issues to date, followed by subsequent issues as they are produced.
3. Members are positively encouraged to contribute news, ideas, articles, letters, problems solutions etc for the newsletter.
4. Queries accompanied by an SAE will receive a personal reply, otherwise the query will be dealt with in the newsletter. No SAE - no personal reply.
5. Letters, routines etc will be assumed to be available for inclusion in the newsletter unless otherwise stated.
6. Depending upon contributions received, PD discs of games, utilities and STOS routines will be available for a small charge, the charge to be shared between club and author where possible.
7. Newsletters will be published in September, November, January, March, May, and July.
8. The helpline is freely available to all members and you are encouraged to offer help too - particularly if you have the ability to patch in machine code etc. If you would like to offer help, write to the club address with brief outlines of your special subject(s).
9. When asking for help it's advisable to ring first as your query may be sorted rapidly on the phone. If more is needed you'll be asked to send further details including discs and printouts where possible. All such material must be accompanied by sufficient return postage/packaging if you want it back!
10. Your wish is my command! (Well maybe!)

LETTERS

As with any club, the STOS user group depends on interaction between members.

So one of the most prominent features of the newsletter will be the letters pages where you can sound off on any subject connected with STOS.

The user group is independent of Mandarin while maintaining close links so affording the best of both worlds, accurate and up to date information, and the chance to voice any moans you may have about STOS.

Mandarin welcome comments on STOS to help them develop further utilities etc. If they know what you want to see you'll have a better chance of getting it!

As this is the first issue of the newsletter, feedback is definitely lacking. What there has been so far follows.....(ignoring the "Enclosed £10 for membership etc...").

CHARLIE FURNESS

I have no pretensions to be a commercial games programmer, but would like to show my grandson I can do something on my Atari that he couldn't do as well on his BBC.

{Too right! Can't let the know-it-alls get the upper hand!}

ANDREW JAMES

Any Ideas how to use ST Replay with STOS? The standard ST Replay code causes a bus error when used!! Plus garbage at

LETTERS

the top of the screen.

{Haven't a clue... this one gets passed on to an expert! Anyone else with a similar interfacing problem? Let me know and I'll see what can be resolved. Please send any data, soft or hardcopy to demonstrate the problem AND return postage for the lot if you want the bumpf back!}

STEVE WATTS

I have just visited the PC show at Earls Court and seen a demonstration of STOS - I just had to buy a copy there and then. I have a little knowledge of Basic programming, but never tried to write games before - so I am hoping STOS will help me.

{Steve... and anyone else with limited experience... don't be afraid to ask for help! STOS is new to us all!}

R TOOHEY

{Terribly formal aren't we! Let's have some first names!} I bought STOS at the PC show - Initial Impressions are good; but had one problem (if you're the person to tell): trying to animate the droid using SPRITE.ACB causes an error (out of memory at line 7015: it tries to reserve more space than is available - on my 520ST anyway). The only way round this (well the easiest) seems to be to rename the floating point module (FLOAT.BIN) so that it isn't loaded; saving 16K, which is enough to see the droid animated.

{This seems to be a problem with early copies of STOS on 520 machines with the new TOS see elsewhere for a solution!}

LETTERS

DENIS PAVETT

I find STOS an excellent product, however I do have one groan about it. I cannot make a run time program despite following the manual word for word ie. load STOSCOPY from the accessory disc, copy files to a blank formatted disc then save the program with a .PRG extension. When trying to load a program from desktop all I get is a TOS ERROR no 35 appear. What can I do?

(Don't panic! The STOSCOPY program you are using was designed to work with an earlier version of the STOS program and doesn't copy all the files in the STOS folder with the newer version. Simply do a manual copy of any files in the STOS folder which haven't been transferred by STOSCOPY. Alternatively you can type in the listing elsewhere in this newsletter which will handle the newer versions. Incidentally, this program can be used to copy ANY folder, not just STOS, but it won't work with folders which have another folder inside.)

CORRESPONDENCE

All correspondence should be sent to :

Pat Winstanley

STOS USER GROUP

STOSCOPY

Versions of STOS from v1.00 to v1.02 do not have a working copy of STOSCOPY. The program is used to make a copy of the complete STOS folder from the master disc to one on which runnable programmes will go.

If you have one of the rogue STOSCOPYs you can either use the GEM desktop to copy the folder or type the following listing in and use as per the STOS manual when STOSCOPY is referred to.

This new version is more versatile then the original STOSCOPY and can be used to copy any folder so long as it does not itself contain a folder. It is versatile and will copy up to 64 files from a folder while working out the memory management.

```

100 rem *****
110 rem * STOSCOPY Version 2.0 by Richard Vanner *
120 rem * (c) 1988 Mandarin/Jawx *
130 rem *****
140 FOLD$="STOS" : drive=0
150 cls : centre "STOSCOPY Version 2.0"
160 print : centre "(c) Mandarin / Jawx 1988" : print : print
170 print : print "Insert a disc containing the ";FOLD$;" folder,
and press a key."
180 wait key
190 dim FILES(64),FILES$(64),FSIZES(64)
200 dir$="\"+FOLD$
210 SOURCE$=dir$+"\*.*"
220 A$=dir first$(SOURCE$,-1) : A$=dir next$

```

STOSCOPY

```

230 A=0
240 repeat
250 FILE$=dir next$ : if FILE$="" then A=-1 : goto 270
260 FILES(A)=1 : FILES$(A)=left$(FILE$,12) :
FSIZES(A)=val(mid$(FILE$,13,8)) : A=A+1
270 until A=-1
280 A=0
290 if FILES(A)=1 then inc A : goto 290 : else NUMFILES=A
300 rem *****
310 rem * Create a new folder onto the new disc *
320 rem *****
330 print : print "Insert a newly formatted disc into the current
drive and press a key." : wait key
340 on error goto 560
350 mk dir FOLD$ : on error goto 0 : print : print "New
";FOLD$;" folder created." : print : print "Now re-insert the
source disc and press a key." : wait key
360 rem *****
370 rem * Copy 15 files or less into the memory banks *
380 rem *****
390 FSTART=0 : FMAX=15
400 NF=FSTART : NUMBANKS=1 : while FILES(NF)=1 and
NUMBANKS<FMAX+1
410 on error goto 550 : reserve as work

```

STOSCOPY

```

NUMBANKS,FSIZES(NF) : on error goto 0
420 print "Loading : ";FILES$(NF);"- Into bank ";NUMBANKS :
bload "\"+FOLD$+"\ "+FILES$(NF),NUMBANKS
430 inc NF : inc NUMBANKS : wend
440 rem *****
450 rem * Save banks out to new folder *
460 rem *****
470 print : print "Insert the destination disc, and press a key" :
wait key
480 NF=0
490 print "Saving : ";FILES$(FSTART);"- from bank ";NF+1 :
b save "\STOS\ "+FILES$(FSTART),start(NF+1) to
start(NF+1)+FSIZES(FSTART)
500 inc FSTART : inc NF : if FSTART=NUMFILES then goto 530
510 if NF<>FMAX then goto 490
520 for A=1 to 15 : erase A : next A : print : print "Insert the
STOS source disc, and press a key." : wait key : FMAX=15 :
goto 400
530 print : print "Copy complete."
540 end
550 FMAX=NUMBANKS-1 : goto 430
560 print : print "I can't open the new folder. Please check the
disc!" : end

```

RAINBOW REVELATIONS

You'll have seen the beautiful demo of a rainbow coloured worm, but how was it done? Let's take the listing to bits and find out.

LINE 40... gets rid of the menu, cursor and mouse, sets the screen to low resolution, and turns off the flash.

Then it sets the palette to be used by giving each colour index a particular value. The indices can be thought of as individual paint pots, numbered 0 to 15, each of which can be filled up with the colour of paint of your choice. Later on, the program will be told which index (or paint pot) to use (dip its brush in) before drawing on the screen.

LINE 50... having filled the indices with different colours, the shift command promptly changes the colours in an orderly way, so that all 512 possible colours get a turn in things. In this case the change will be made every twenty-fifth of a second and because it runs under interrupt will continue to do so until deliberately stopped.

LINE 60... this draws a line from the centre of the screen to the top left hand corner, then from the centre to a point on the top line two pixels to the right of the last until the last line is drawn from the centre to the top right hand corner of the screen. The variable X tells the program where on the top line the drawn line should go.

Meanwhile the variable Y is used to control the ink used to draw with. The expression $Y \text{ MOD } 15$ simply takes the current value of Y (which is increased by one for every revolution of the loop), divides it by 15 and returns the remainder. Thus if Y had a value of 63 then $Y \text{ MOD } 15$ would return a value of 3. Adding one to this gives 4, so in this case the line to be drawn would be shown in the colour currently in index 4, whatever that happened to be.

RAINBOW REVELATIONS

This colour however is changing rapidly and constantly due to the SHIFT command earlier.

LINES 70-90... do the same as line 60, except that line 70 draws to points on the right edge of the screen, line 80 right to left along the bottom, and line 90 bottom to top on the left edge of the screen.

Once all the lines have been drawn they remain in place but appear to move because of the SHIFT command earlier which is regularly changing the colours of the lines.

LINE 110... having set up the background of lines, the next step is to draw the snake. If you watch the program in action you can see that the circles are placed progressively along a sine wave. To achieve this the x co-ordinate of the centre of the circle is moved 4 pixels to the right after each circle is printed. Meanwhile the y co-ordinate is moved up or down depending on the value of the expression $\sin((X/320.0)*2*\pi)*75+100$. (If you have enough maths you should be able to see how this works - if not, don't worry about it - just try substituting different numbers in the expression and see what happens. You should be able to adjust the loops of the wave so that they become flatter or more peaked.)

LINE 120... this line simply takes up where the previous one left off, and draws an identical pattern from right to left across the screen. Once the left edge has been reached all the circle printing has been done and as with the lines earlier, the picture itself remains static while the rotation of colours gives the impression of movement. After a while (wait 150) the demo terminates itself by fading all the colours in the indices to white, then line 130 stops the colour shift and fades all the colours to black.

Line 140 simply resets the editor and puts you back into it.

RAINBOW REVELATIONS

10 rem *****

20 rem * Demo of shifting colours *

30 rem *****

40 key off : curs off : hide : mode 0 : flash off : palette
\$0,\$702,\$700,\$730,\$750,\$770,\$470,\$70,\$75,\$77,\$57,\$27,\$7,\$507,
\$707,\$704

50 shift 2

60 Y=0 : for X=0 to 319 step 2 : inc Y : ink (Y mod 15)+1 : draw
160,100 to X,0 : next X

70 for X=0 to 199 step 2 : inc Y : ink (Y mod 15)+1 : draw
160,100 to 319,X : next X

80 for X=319 to 0 step-2 : inc Y : ink (Y mod 15)+1 : draw
160,100 to X,199 : next X

90 for X=199 to 0 step-2 : inc Y : ink (Y mod 15)+1 : draw
160,100 to 0,X : next X

100 T=100

110 Z=0 : for X=0 to 319 step 4 : Y=sin((X/320.0)*2*pi)*75+100 :
inc Z : ink (Z mod 15+1) : circle X,Y,25 : next X

120 for X=319 to 0 step-4 : Y=-sin((X/320.0)*2*pi)*75+100 : inc
Z : ink (Z mod 15+1) : circle X,Y,25 : next X : wait 150 : fade
2,\$777,\$777,\$777,\$777,\$777,\$777,\$777,\$777,\$777,\$777,\$777,
7,\$777,\$ 777 , \$777,\$777

130 wait 14 : shift off : fade 2 : wait 14

140 default