ARTIFACTING

How would you like to see a new set of colours on your Atari? What about a true red and new vibrant greens, yellows, and purples? And how about an almost endless combination of multicolour characters in graphics 0? And all this without using player-missile graphics or display list interrupts. Impossible? Type in the program and be amazed.

It uses a phenomenon known as 'artifacting' previously only used on American programs. This consists of plotting pixels in special striped or checkered patterns which, because of the way colour is stored in a television signal, produces different colours than those normally associated with the pixels.

This technique was described by various articles which appeared in US Atari magazines and has been used in some commercial programs, such as the Ultima series by Origin Systems Inc. Unfortunately, due to the differences in the American NTSC and our PAL television systems, any British Atari owner would see dull grey stripes tinted with the occasional horrible blue or green.

A few years ago, I read an article by Anthony Ball, explaining how he discovered some PAL artifacting ability. The programs he presented did show some startling new colour in the high resolution graphics 8 mode. Since then, I have been surprised that nobody has written any programs or articles to take advantage of these colours never before seen on the Atari. I decided it was time to rectify this and this article is a guide to PAL artifacting as a result of my own 'experiments'.

Artifacting can be used on either graphics modes 0 or 8. These use the highest resolution available on the 8-bit Atari. Rather than try to plot the special patterns required on a graphics 8 screen, which is very difficult, the program uses graphics 0 characters, the artifacted colours being easier to produce and alter.

Diagram 1 below shows the character patterns for the new yellow and violet. These patterns are exactly the same except one is shifted four pixels over from the other. The vertical stripes cause the artifacting to occur and the movement by four pixels changes the colour from yellow to violet. With the more complex checkered patterns in diagram 2, moving the red character up or down by one pixel will give the green.

Joel Goodwin explains how to use artifacting on the PAL system for true additional colours

HOW TO USE ARTIFACTING IN YOUR PROGRAMS

As with everything, nothing is perfect and there are some conditions to be met if you are to be able to use and see these wonderful new colours. Firstly, the colour on your television needs to be turned up high. You will be reminded of this at the beginning of the program. The second condition is that the screen needs to be adjusted slightly for the correct colours to appear every time and I have written a routine to do this.

The routine is in lines 15000-15290. Simply include it in your own program and follow these instructions. First, once your character set is redefined, place the location of the character set in the variable FONT and insert a GOSUB 15200 command. This will run the routine to adjust the screen. The user will be asked to press keys 1-8 until the red, green and yellow colours displayed match the words underneath them. Only one of these keys will be the correct adjustment and it is usually one of the first four. If you have used different characters for the red, green or yellow colours, adjust line 15200 accordingly, which contains these characters in that order. Also, avoid using the rightmost column as the routine may scroll the screen to the right by 1 to 3 pixels causing anything printed in this column to overlap on the left side of the screen. Whenever you need to issue a GRAPHICS 0 command use GOSUB 15000 instead. Note, the routine uses locations 1536-1613 of page six.

THE EDITOR PROGRAM

When you run the program, you will first see the adjustment screen. Follow the onscreen instructions and then press the Space Bar. You will then be given a choice of seeing a demo screen or going to the editor. The demo shows something of what can be achieved using combinations of these colours. The main program is a character set editor dedicated to artifacting (you can also use it to design ordinary characters). Artifacting can be produced on other character editors, but without the screen adjustment routine, you would need to keep rebooting until the colours appeared correctly.

The main screen of the program displays the 64 multicolour patterns and colours that I have discovered so far. Each of them is displayed in 4-character blocks on the left of the screen. On the right is an 8x8 grid for editing. You can make use of any of these patterns in your programs or use the editor to alter them and create even more. Just use my patterns as a

```
EW 0 REM ##############################
UZ 1 REM #         ARTIFACTING          #
VA 2 REM #      CHARACTER EDITOR        #
GE 3 REM #              by              #
JL 4 REM #         Joel Goodwin         #
HC 5 REM # -------------------------- #
TL 6 REM # NEW ATARI USER - OCT 1991  #
FD 7 REM ##############################
NN 8 REM
LQ 10 GOTO 100
RL 19 REM  DRAW CHARACTER GRID
IY 20 MEM=FONT*256+CH*8:Z=0:CL=C1
ZW 30 FOR ROW=Y TO Y+7:VALUE=128:BYTE=PEE
   K(MEM+ROW-Y):IF CHK THEN Z=(Z=0)
TK 40 IF BYTE=0 AND CHK=0 THEN COLOR CL:P
   LOT X,ROW:DRAWTO X+7,ROW:NEXT ROW:RETU
   RN
PI 50 FOR BIT=X TO X+7:IF CHK THEN Z=(Z=0
   ):CL=C1*Z+C2*(1-Z)
KB 60 IF BYTE<VALUE THEN COLOR CL:GOTO 90
JO 70 BYTE=BYTE-VALUE:COLOR C3:IF VALUE=1
    OR BYTE>0 OR CHK THEN 90
FI 80 PLOT BIT,ROW:COLOR CL:PLOT BIT+1,RO
   W:DRAWTO X+7,ROW:BIT=50:GOTO 95
PS 90 PLOT BIT,ROW
VX 95 VALUE=VALUE/2:NEXT BIT:NEXT ROW:RET
   URN
HJ 100 FONT=PEEK(106)-4:POKE 106,FONT-1:G
   RAPHICS 0:GOSUB 1000
PB 110 DRAW=20:IF  NOT SKIP THEN GOSUB 33
   00
HP 120 GOSUB 3000:SKIP=0
XF 199 REM  MAIN LOOP
DB 200 GOSUB DRAW:GOSUB 570:Z=0:POKE POS,
```

```
   CUR:Z=0:POKE 556,1:POKE 542,W:Z=0
QX 210 S=PEEK(632):T=PEEK(53264):IF PEEK(
   764)<>255 THEN 600
XT 220 IF  NOT T THEN 350
PN 230 IF PEEK(556) OR S<>15 THEN 260
QY 240 Z=(Z=0):POKE POS,CS*Z+CUR*(Z=0)
JQ 250 POKE 556,1:POKE 542,W:GOTO 210
MX 260 IF S=15 THEN 210
UY 270 DX=J(S,0):DY=J(S,1)
IC 280 X1=CX+DX:IF X1<0 OR X1>38 THEN X1=
   CX
FY 290 Y1=CY+DY:IF Y1<0 OR Y1>22 THEN Y1=
   CY
JA 300 IF X1=CX AND Y1=CY THEN 210
EC 310 POKE POS,CS:CX=X1:CY=Y1
GB 320 POS=SC+CX+CY*40:CS=PEEK(POS):CUR=6
   2+(CS>127)*128
QJ 330 POKE POS,CUR:Z=0:POKE 542,W:POKE 5
   56,1:GOTO 210
WL 349 REM  TRIGGER SUBROUTINE
SB 350 IF CS=0 THEN 210
XX 360 IF JMP THEN GOSUB 850
ZD 370 POKE 53279,1:IF CX>24 THEN 390
JF 380 CH=CS:GOTO 200
XL 390 IF CS<>128 AND CS<>63 THEN 210
PJ 400 IF CH=0 THEN 230
GI 410 CS=128-(CS=128)*65:CUR=62+128*(CS>
   127)
XN 420 BIT=8-CX+X:ROW=CY-Y:VALUE=2^BIT/2:
   BYTE=PEEK(MEM+ROW)
KT 430 OP=SGN(CS-127):POKE MEM+ROW,BYTE+O
   P*VALUE
QZ 440 POKE POS,CUR:S=STICK(0):GOTO 260
NW 569 REM  SCREEN RESET
```

starting point.

The flashing circle on the screen is the cursor. Use the joystick in port 1 to move it. Press the trigger when over a character and it's enlarged pattern will be displayed on the grid. To see the character data of this pattern, press the Space Bar. Above the grid more copies of the selected character will be displayed, 16 in a block and 2 separately. These will mirror any changes made on the grid and make it easy for you to see what effect your editing is having. You can alter the pattern by moving the cursor onto the grid and pressing the trigger to draw or erase. The arrow keys will allow you to scroll the whole character in any direction. Each keypress moves it by one pixel and this can be done as often as you like. When you are satisfied just move off the grid and choose another pattern to alter.

Using the scrolling can give quick results as there are many different positions possible for each character. Not all of these positions give useful colours and some give repeat colours, but it is fun to experiment and see what you can discover. To create multicolour characters all you need do is combine a checkered and striped pattern and scroll it around until you find one you like. Just drawing at random and scrolling can quickly create something new. Bare in mind that two pixels side-by-side in any row will produce white.

The program understands the following commands.

ESC	- clears character
R	- restores original character
ARROW KEYS	- scroll up down left and right
SPACE BAR	- shows character data
S	- saves out modified character set to cassette or disk for storage
L	- loads in character set from cassette or disk
H	- brings up a help screen with all these commands displayed

CONTINUED ☞

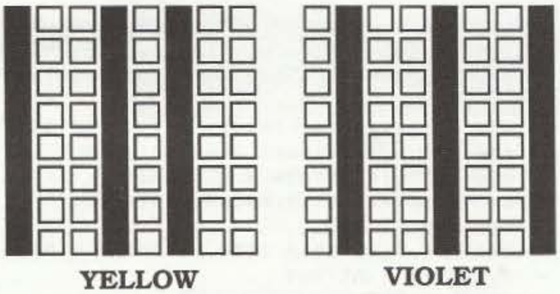

Diagram 1

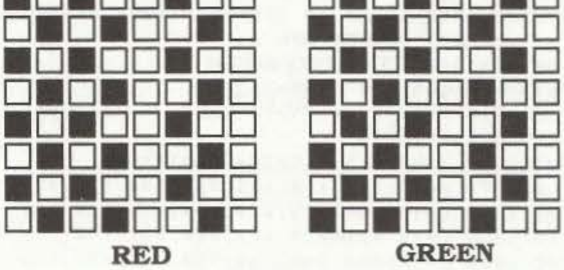

Diagram 2

COMING NEXT ISSUE ...

RUNAROUND - a game writing with artifacting

```
BO 570 Z=PEEK(POS):IF Z<>CS AND Z<>CUR TH
   EN CS=Z:CUR=62+(CS>127)*128
OO 575 IF SKIP THEN SKIP=0:RETURN
MM 580 POSITION 27,20:? #6;"PRESS H":POSI
   TION 27,21:? #6;"FOR HELP":IF  NOT CH
   THEN RETURN
RK 590 POKE SC+269,CH:POKE SC+272,CH:FOR
   P=1 TO 4:FOR Z=29 TO 32:POKE SC+Z+P*40
   ,CH:NEXT Z:NEXT P:RETURN
WQ 599 REM  KEYBOARD ROUTINE 
YF 600 RESTORE :K=PEEK(764):IF  NOT CH TH
   EN IF K<>57 AND K<>0 THEN POKE 764,255
   :GOTO 210
VS 610 READ P,Z:IF P=-1 THEN POKE 764,255
   :GOTO 210
DD 620 IF K<>P THEN 610
SN 625 IF JMP THEN 850
XI 630 GOSUB 900:GOSUB Z:POKE 764,255:SKI
   P=1:GOTO 200
RX 640 DATA 6,650,7,650,14,770,15,750,33,
   800,28,950,40,970,57,3200,0,3400,62,34
   00
AS 649 REM  HORIZONTAL SCROLLING 
UO 650 OP=2:IF K=7 THEN OP=0.5
LJ 660 FOR ROW=0 TO 7
NI 670 BYTE=PEEK(MEM+ROW):BYTE=BYTE*OP
VV 680 IF BYTE<>INT(BYTE) THEN BYTE=BYTE+
   127.5
OT 690 IF BYTE>255 THEN BYTE=BYTE-255
MG 700 POKE MEM+ROW,BYTE:NEXT ROW:RETURN
XM 749 REM  VERTICAL SCROLLING 
DE 750 P=PEEK(MEM+7):FOR ROW=7 TO 1 STEP
   -1
ZB 760 POKE MEM+ROW,PEEK(MEM+ROW-1):NEXT
   ROW:POKE MEM,P:RETURN
MN 770 P=PEEK(MEM):FOR ROW=0 TO 7
IE 780 POKE MEM+ROW,PEEK(MEM+ROW+1):NEXT
   ROW:POKE MEM+7,P:RETURN
XX 799 REM  DISPLAY DATA 
EZ 800 D$="DATA "
LH 810 FOR ROW=0 TO 7:N$=STR$(PEEK(MEM+RO
   W)):IF ROW<7 THEN N$(LEN(N$)+1)=","
EX 820 P=LEN(D$):D$(P+1,P+LEN(N$))=N$:NEX
   T ROW
HJ 830 POSITION 19-LEN(D$)/2,23:? #6;D$;
LI 840 POP :JMP=1:POKE 764,255:GOTO 210
MF 850 JMP=0:POSITION 2,23:D$(1)=" ":D$(3
   6)=" ":D$(2)=D$:? #6;D$;:GOTO 210
CT 899 REM  BEEP 
TP 900 FOR P=15 TO 0 STEP -5:SOUND 0,50,1
   0,P:NEXT P:RETURN
YQ 949 REM  CLEAR CHARACTER 
KI 950 FOR ROW=0 TO 7:POKE MEM+ROW,0:NEXT
    ROW:RETURN
FU 969 REM  RESTORE CHARACTER 
FX 970 RESTORE 1360+CH*10
TH 980 FOR ROW=0 TO 7:READ P:POKE MEM+ROW
   ,P:NEXT ROW:RETURN
XU 999 REM  CHARACTER SET 
OI 1000 RESTORE 1000:J=1535
XF 1010 READ Z:IF Z<>-1 THEN J=J+1:POKE J
   ,Z:GOTO 1010
ZC 1020 Z=USR(1536,57344,FONT*256)
EU 1030 DATA 104,104,133,213,104,133,212
NT 1040 DATA 104,133,215,104,133,214,162
JP 1050 DATA 4,160,0,177,212,145,214
OS 1060 DATA 200,208,249,230,213,230,215
ZU 1070 DATA 202,208,240,96,-1
HO 1080 X=488:Y=567:GOSUB 1150
LD 1090 GOSUB 15200:GOSUB 1200
GT 1100 GOSUB 15000:POSITION 14,12:? "PLE
   ASE WAIT"
AY 1110 X=568:Y=630:GOSUB 1150
QT 1120 POKE 559,0
BU 1130 X=631:Y=1023:GOSUB 1150
AN 1140 RETURN
FH 1150 FOR Z=FONT*256+X TO FONT*256+Y
```

```
CV 1160 READ C:POKE Z,C:NEXT Z:RETURN
MH 1200 ? CHR$(125):POSITION 7,8:? "PRESS
    1 FOR DEMO SCREEN":POSITION 10,9:? "O
   R 2 FOR EDITOR"
EU 1210 OPEN #1,4,0,"K:"
TR 1220 GET #1,K:K=K-49:IF K<>0 AND K<>1
   THEN 1220
XJ 1230 CLOSE #1:SKIP=K:? CHR$(125):RETUR
   N
AQ 1970 DATA 0,124,16,16,16,16,124,0
SZ 1980 DATA 0,60,126,102,102,126,60,0
UY 1990 DATA 0,126,66,66,66,66,126,0
OG 2000 DATA 41,82,41,82,41,82,41,82
OX 2010 DATA 164,82,164,82,164,82,164,82
OO 2020 DATA 82,164,82,164,82,164,82,164
IH 2030 DATA 74,37,74,37,74,37,74,37
EO 2040 DATA 73,73,73,73,73,73,73,73
VH 2050 DATA 148,148,148,148,148,148,148,
   148
HZ 2060 DATA 8,16,36,130,65,36,8,16
XT 2070 DATA 32,80,160,0,0,82,5,2
OL 2080 DATA 33,66,33,16,8,66,33,66
ZQ 2090 DATA 164,66,164,16,0,66,164,82
AS 2100 DATA 66,164,66,0,16,164,66,164
IC 2110 DATA 66,37,66,0,8,37,66,37
SU 2120 DATA 5,144,32,64,32,64,144,10
KV 2130 DATA 0,16,40,69,40,16,0,16
MY 2140 DATA 84,148,146,130,52,18,148,84
WI 2150 DATA 16,40,16,10,5,2,4,8
ZG 2160 DATA 21,82,16,32,16,8,20,82
VM 2170 DATA 0,0,16,36,0,82,164,66
QC 2180 DATA 2,5,136,82,37,136,80,32
WZ 2190 DATA 148,72,136,73,148,84,148,73
GO 2200 DATA 0,82,40,82,40,82,40,0
UB 2210 DATA 0,82,36,82,36,82,36,0
UN 2220 DATA 0,36,82,36,82,36,82,0
XK 2230 DATA 0,36,74,36,74,36,74,0
CG 2240 DATA 168,21,168,21,168,21,168,21
KZ 2250 DATA 42,69,42,69,42,69,42,69
QO 2260 DATA 162,84,162,84,162,84,162,84
IH 2270 DATA 21,168,21,168,21,168,21,168
PQ 2280 DATA 82,164,82,164,164,82,164,82
OH 2290 DATA 41,82,41,82,82,41,82,41
LA 2300 DATA 82,41,82,82,41,82,41,41
JE 2310 DATA 164,82,164,73,164,82,164,164
KQ 2320 DATA 170,65,170,73,170,73,162,85
WO 2330 DATA 16,40,84,84,84,84,40,16
IC 2340 DATA 20,82,21,82,0,0,0,0
SH 2350 DATA 10,72,72,33,33,207,72,20
QX 2360 DATA 160,18,36,16,160,2,4,2
AY 2370 DATA 2,160,18,132,16,160,2,132
OH 2380 DATA 164,144,160,130,132,18,36,14
   6
ZP 2390 DATA 74,33,66,9,10,40,72,41
YJ 2400 DATA 20,18,22,84,152,34,196,18
AG 2410 DATA 82,0,164,82,180,82,0,164
RB 2420 DATA 41,0,82,57,82,41,0,82
IN 2430 DATA 82,0,41,90,41,82,0,41
AS 2440 DATA 164,0,82,180,82,164,0,82
QR 2450 DATA 144,144,148,148,148,148,20,2
   0
HI 2460 DATA 148,0,148,148,180,148,0,148
MJ 2470 DATA 73,73,74,148,74,148,73,73
YU 2480 DATA 41,42,41,42,41,42,41,40
QF 2490 DATA 82,85,82,85,82,85,82,85
RH 2500 DATA 164,21,164,21,164,21,164,164
EN 2510 DATA 8,36,74,164,82,37,74,36
CH 2520 DATA 37,82,84,82,37,34,36,82
HR 2530 DATA 36,18,36,18,0,36,18,36
CO 2540 DATA 40,82,8,32,82,36,82,8
VB 2550 DATA 0,0,74,160,64,4,10,164
SG 2560 DATA 0,0,64,168,168,84,138,36
VO 2570 DATA 0,66,5,18,0,72,160,66
UN 2580 DATA 20,42,85,170,85,170,84,40
FR 2590 DATA 73,170,136,73,73,170,73,170
CV 2600 DATA 255,211,169,211,169,211,169,
   255
```

```
PO 2610 DATA 255,211,165,211,165,211,165,
   255
LZ 2620 DATA 255,165,211,165,211,165,211,
   255
PO 2630 DATA 255,165,203,165,203,165,203,
   255
LQ 2999 REM  INITIALISE DISPLAY 
ZI 3000 X=27:Y=8:CX=31:CY=16:C1=95:C3=160
   :CUR=62:W=10:DIM D$(36),N$(4),C$(9),FN
   $(14):RESTORE 3000
QG 3010 GOSUB 15000:POKE 16,64:POKE 53774
   ,54:SC=PEEK(88)+PEEK(89)*256:C$="CHARA
   CTER":POKE 82,26
FF 3020 DX=1:DY=0:FOR CH=64 TO 127
YC 3030 FOR P=0 TO 1
RG 3040 POKE SC+DX+P+DY*40,CH
AQ 3050 POKE SC+DX+P+40+DY*40,CH
EW 3060 NEXT P:DX=DX+3:IF DX=25 THEN DX=1
   :DY=DY+3
BM 3070 NEXT CH:CH=0:POS=SC+CX+CY*40
GZ 3080 DIM J(14,1)
LC 3090 FOR P=5 TO 14
LN 3100 READ DX,DY
TK 3110 J(P,0)=DX:J(P,1)=DY
JG 3120 NEXT P:RETURN
YS 3130 DATA 1,1,1,-1,1,0,0,0,-1,1,-1,-1,
   -1,0,0,0,0,1,0,-1
EX 3199 REM  HELP SCREEN 
DJ 3200 GOSUB 3290
GF 3210 POSITION 26,1
ZN 3220 ? #6;"ESC CLEAR":? #6;C$:? #6;"R
   RESTORE":? #6;C$:? #6
JV 3230 ? #6;"ARROW KEYS":? #6;"SCROLL":?
    #6;C$:? #6;"UP,DOWN,":? #6;"LEFT,RIGH
   T"
GH 3240 ? #6;"SPACE SHOWS":? #6;C$:? #6;"
   DATA":? #6
XY 3250 ? #6;"S SAVES":? #6;"MODIFIED":?
   #6;C$;" SET":? #6;"L LOADS":? #6;C$;"
   SET"
BU 3260 ? #6:? #6;"TRIGGER TO":? #6;"CONT
   INUE"
XQ 3270 IF STRIG(0) THEN 3270
RR 3280 GOSUB 3290:POKE 764,255:POP :GOTO
    200
KU 3290 COLOR 32:FOR C=25 TO 38:PLOT C,1:
   DRAWTO C,22:NEXT C:RETURN
DG 3299 REM  DEMO SCREEN 
OO 3300 ? #6;CHR$(125):RESTORE 3300:CHK=1
   :C3=160:X=32:Y=1
QA 3310 READ C1,C2,CH:IF C1=-1 THEN 3340
QZ 3320 GOSUB DRAW:X=X-8:IF X=-8 THEN COL
   OR 32:PLOT 39,1:DRAWTO 39,8:X=27:Y=10
PY 3330 GOTO 3310
BV 3340 POKE 764,255:POSITION 6,21:? #6;"
   PRESS SPACE BAR FOR EDITOR":POKE 559,3
   4
FP 3350 IF PEEK(764)<>33 THEN 3350
VK 3360 CHK=0:GOSUB 900:POKE 764,255:RETU
   RN
HM 3370 DATA 118,103,61,9,106,50,252,120,
   33,114,14,52,1,2,33
YY 3380 DATA 119,14,50,99,20,37,100,100,5
   1,120,8,53,-1,-1,-1
VV 3399 REM  LOAD AND SAVE CHAR. SET 
YG 3400 POP :P=(K=62)*4+4:POKE 764,255
DQ 3410 GOSUB 3290
TS 3420 POSITION 26,8:IF P=8 THEN 3500
VA 3430 ? #6;"LOAD":GOSUB 3550
CI 3440 IF FN$="C:" THEN ? #6;"."
WP 3450 ? #6:? #6;"TO LOAD,":? #6;"PRESS
   RETURN.":GOSUB 3480
QO 3460 GET #1,Z:POKE C+BYTE,Z:BYTE=BYTE+
   1:IF BYTE<1024 THEN 3460
BH 3470 CLOSE #1:CLOSE #2:GOSUB 3290:GOTO
    200
```

```
OI 3480 TRAP 3620:BYTE=0:C=FONT*256:IF FN
   $<>"C:" THEN GET #2,K
JH 3490 OPEN #1,P,0,FN$:RETURN
CW 3500 ? #6;"SAVE":GOSUB 3550
IG 3510 IF FN$="C:" THEN ? #6:? #6;"AND R
   ECORD."
JB 3520 ? #6:? #6;"TO SAVE,":? #6;"PRESS
   RETURN.":GOSUB 3480
BD 3530 Z=PEEK(C+BYTE):PUT #1,Z:BYTE=BYTE
   +1:IF BYTE<1024 THEN 3530
TR 3540 GOTO 3470
AL 3550 ? #6:? #6;"C CASSETTE":? #6;"D DI
   SK":? #6:OPEN #2,4,0,"K:":GET #2,K
WA 3560 IF K<>68 AND K<>67 THEN POP :GOTO
    3470
AU 3570 IF K=67 THEN FN$="C:":GOTO 3600
GB 3580 ? #6:? #6;"FILENAME?":POSITION 25
   ,15:? #6;"D:";:INPUT #16;D$:IF D$="" T
   HEN 3470
UB 3590 FN$="D:":FN$(3)=D$:? #6
OH 3600 ? #6;"INSERT ";:IF FN$="C:" THEN
   ? #6;"TAPE &":? #6;"PRESS PLAY";:RETUR
   N
EW 3610 ? #6;"DISK.":RETURN
TN 3620 GOSUB 3290:GOSUB 900:GOSUB 900:CL
   OSE #1
JS 3630 POSITION 26,10:? #6;"ERROR ";PEEK
   (195)
VZ 3640 ? #6:? #6;"PRESS ANY KEY":GET #2,
   K:CLOSE #2:GOSUB 3290:GOTO 200
AK 14995 REM  SCREEN ADJUSTMENT ROUTINE 
SQ 14996 REM USE 15000-15290 IN YOUR OWN
FQ 14997 REM PROGRAMS.
FJ 14998 REM
QJ 14999 REM  CREATE ADJUSTABLE GR.0 
XX 15000 GRAPHICS 0:POKE 710,0:POKE 752,1
   :? CHR$(125);:POKE 756,FONT:XSC=PEEK(8
   8)+PEEK(89)*256:XSC=XSC-4
TY 15010 POKE XSC+3,0
VY 15020 POKE 1536,112-(XVT*16):POKE 1537
   ,112:POKE 1538,112
TR 15030 FOR XD=0 TO 23:POKE 1539+XD*3,82
   :XSHI=INT(XSC/256):XSLO=XSC-XSHI*256
ZC 15040 POKE 1540+XD*3,XSLO:POKE 1541+XD
   *3,XSHI
PM 15050 XSC=XSC+40:NEXT XD
YC 15060 POKE 1539+XD*3,65:POKE 1540+XD*3
   ,0:POKE 1541+XD*3,6
WU 15070 POKE 54276,XHZ:POKE 560,0:POKE 5
   61,6:POKE 709,10:RETURN
YP 15099 REM  ADJUSTING ROUTINE 
AD 15100 XVT=INT(XADJ/5):XHZ=XADJ-XVT*4-1
EW 15110 POKE 54276,XHZ:POKE 1536,112-(XV
   T*16)
DS 15120 RETURN
FX 15199 REM  COLOUR ADJUSTMENT SCREEN 
LV 15200 GOSUB 15000:POKE 752,1:? :FOR B=
   1 TO 3:POSITION 5,9+B:? "├├├├├├├├├ |||
   |||||| ┐┐┐┐┐┐┐┐┐":NEXT B
AV 15210 ? "     Red      Green      Yello
   w  "
FS 15220 POSITION 2,19:? "PRESS NUMBERS F
   ROM 1 TO 8 UNTIL ALL":? "  THREE COLOU
   RS ABOVE ARE CORRECT"
SK 15230 ? "      THEN PRESS THE SPACE BAR
   .":POSITION 2,2:? "THIS PROGRAM USES A
   RTIFACTED COLOURS"
NW 15240 ? " TO SEE THEM PROPERLY YOU MUS
   T TURN":POSITION 8,4:? "YOUR TV COLOUR
    UP HIGH!"
ZG 15250 OPEN #5,4,0,"K:"
DF 15260 GET #5,XK:IF XK=32 THEN CLOSE #5
   :RETURN
AW 15270 XK=XK-48:IF XK<1 OR XK>8 THEN 15
   260
HP 15280 XADJ=XK:GOSUB 15100
EH 15290 GOTO 15260
```

●