

APAC Graphics Mode#

General Information

Author: THOMAS TANIDA

Published: 1987

Demo of APAC Graphics Mode#

Any Point, Any Color

The APAC System, or Any Point, Any Color was a software-driven display mode capable of displaying an image using all 256 of the Atari's possible colors. By taking 80×192 mode lines that displayed 16 hues, and those that displayed 16 shades, and either interlacing rows of them, quickly alternating between rows of them, or both, a screen displaying 80×96 or 80×192 pixels in 256 colors could be perceived.

APAC was created in early 1987 and later introduced in the magazine A.N.A.L.O.G. Computing, Issue #60, May 1988 in an article by Tom Tanida. The source code was written in 6502 assembly language.

APAC used a Display List Interrupt, or DLI, after each line of the screen was drawn to alternate between GTIA Graphics Mode 9 (15 hues) and 11 (15 shades of grey) of the GTIA chip. The hues and luminances would blend together on the screen (usually a television) to create the effect of a palette of 256 visible colors, with the artifact of a thinner, horizontal blank line in between each visible line.

APAC used a very basic API consisting of four functions:

- Init, used to place the computer into the APAC mode
- Exit, used to exit the APAC mode
- Plot, used to place a point of a specified color on the screen
- Draw, used to draw a line between the last plotted point to the given point

A second article for an "APAC-II" mode was hinted at in the original article. This mode would have alternated the GTIA 9 and 11 modes during a vertical blank interrupt, or VBI. The article was neither completed nor published.

APAC Graphics Mode#

```
10 *****
20 * APAC SYSTEM, V1.1          *
30 * CREATED BY THOMAS TANIDA  *
40 * FIRST STARTED: 1/3/87     *
50 * LAST REVISED: 2/7/87     *
60 *****
70      *= $22FD
80      .OPT NO EJECT
90      .TAB 8,12,30
0100 *****
0110 ZTEMP = $00
0120 COLSAV = 203
0130 YBYT = 204
0140 T8 = 206
0150 FX = 212      ;212-222 USED
0160 FY = 213      ; FOR DRAWTO
0170 XD = 214
```

```

0180 YD = 215
0190 XACC = 216
0200 YACC = 217
0210 DELTX = 218
0220 DELTY = 219
0230 EPOINT = 220
0240 TMP1 = 221
0250 COUNT = 222
0260 *****
0270 *          SYSTEM EQUATES          *
0280 *****
0290 CASINI = $02      ;2
0300 TRAMSZ = $06      ;6
0310 WARMST = $08      ;8
0320 DOSVEC = $0A      ;10
0330 DOSINI = $0C      ;12
0340 ROWCRS = $54      ;84
0350 COLCRS = $55      ;85
0360 SCREEN = $58      ;88
0370 RAMTOP = $6A      ;106
0380 LOMEM = $80       ;128
0390 VDSLST = $0200    ;512
0400 VVBLKI = $0222    ;546
0410 SDMCTL = $022F    ;559
0420 SDLSTL = $0230    ;560
0430 GPRIOR = $026F    ;623
0440 COLOR4 = $02CB    ;712
0450 MEMLO = $02E7     ;743
0460 ICCOM = $0342     ;834
0470 ICBADR = $0344     ;836
0480 ICBLN = $0348     ;840
0490 ICAUX1 = $034A     ;842
0500 COLBK = $D01A     ;53274
0510 PRIOR = $D01B     ;53275
0520 WSYNC = $D40A     ;54282
0530 VCOUNT = $D40B     ;54283
0540 NMIEN = $D40E     ;54286
0550 CIOV = $E456      ;58454
0560 SETVBV = $E45C     ;58460
0570 SYSVBV = $E45F     ;58463
0580 *****
0590 * (FOR BINARY LOADS)
0600     JMP SYSSET
0610 *
0620 * BASIC ENTRY POINTS:
0630     JMP BASPLT      ;PLOT
0640     JMP BASDRW2     ;DRAWTO
0650     JMP BASEXGT     ;EXIT APAC
0660     JMP BASINGT     ;INIT APAC
0670 * ML ENTRY POINTS:
0680     JMP PLT256      ;PLOT
0690     JMP DRT256      ;DRAWTO
0700     JMP EXITGT      ;EXIT APAC
0710     JMP INITGT      ;INIT APAC
0720 *****
0730 * BASIC ENTRY:
0740 * Q=USR(BASPLT,X,Y,COL)
0750 *
0760 * THIS SUBR PLOTS A POINT

```

```

0770 * ON THE APAC SCREEN AT THE
0780 * GIVEN X,Y POSITION USING THE
0790 * GIVEN COLOR (0-255).
0800 * REMEMBER: THERE ARE 192 LINES,
0810 * ALTERNATING LUM AND COL,
0820 * SO THE SCREEN LIMITS ARE
0830 * 80 HORIZONTAL AND 192/2=96
0840 * VERTICAL
0850 *
0860 * FIRST MAKE SURE THERE ARE 3
0870 * (NO MORE, NO LESS) ARGUMENTS
0880 *
0890 BASPLT
0900     PLA
0910     CMP #3
0920     BEQ GETPARM
0930     JMP TRAP      ;WRONG # OF ARGS
0940 GETPARM
0950     PLA           ;IGNORE HI-BYTE
0960     PLA
0970     TAX           ;X-POS
0980     PLA           ;IGNORE HI
0990     PLA
1000     TAY           ;Y-POS
1010     PLA
1020     PLA           ;COLOR
1030 * CHECK FOR ILLEGAL PLOT
1040     JSR BOUNDCK
1050 * ML ENTRY: A=COLOR
1060 *           X=X-POS
1070 *           Y=Y-POS
1080 PLT256
1090     STA COLSAV    ;SAVE COLOR
1100 PSAMCOL
1110     STX COLCRS    ;UPDATE CURSOR
1120     STY ROWCRS
1130 * GET LO-BYTE OF SCREEN MEM PNTR
1140     LDA SCRALO,Y
1150     STA YBYT
1160 * GET HI-BYTE OF SCREEN MEM PNTR
1170     LDA SCRAHI,Y
1180     STA YBYT+1
1190     TXA
1200     LSR A         ;A=A/2
1210     TAY
1220 * GET BYTE WITH PIXEL FROM THE
1230 * SCREEN CONTAINING THE LUM
1240     LDA (YBYT),Y
1250     STA T8         ;SAVE IT
1260     TXA
1270     AND #1         ;X=1=ODD X-POS
1280     TAX           ;X=0=EVEN
1290     LDA T8
1300 * MASK PREVIOUS PIXEL FROM SCRIN
1310     AND BITABL,X
1320     STA T8
1330 * RETRIEVE COLOR
1340     LDA COLSAV
1350     CPX #1

```

```

1360     BNE LHNYB     ;X IS EVEN
1370 * THIS WORKS SINCE THE LUM
1380 * IS IN THE LOWER NYBBLE OF THE
1390 * COLOR, AND IF X IS ODD THEN
1400 * THE LOWER NYBBLE OF THE SCR
1410 * BYTE WILL CONTAIN THE PIXEL
1420     AND #$0F
1430     JMP POKLUM
1440 * MOVE LUM INTO HI-NYBBLE
1450 LHNYB
1460     ASL A
1470     ASL A
1480     ASL A
1490     ASL A
1500 * POKE THE LUM BACK INTO THE SCR
1510 POKLUM
1520     ORA T8
1530     STA (YBYT),Y
1540     CLC
1550     LDA YBYT
1560     ADC #40         ;ADD 40 FOR THE
1570     STA YBYT       ;NXT LINE (COLR)
1580     BCC GETCOLR
1590     INC YBYT+1
1600 * GETS THE BYTE CONTAINING THE
1610 * PIXEL WITH THE COLOR AND SAVE
1620 * IN T8
1630 GETCOLR
1640     LDA (YBYT),Y
1650     AND BITABL,X
1660     STA T8
1670     LDA COLSAV
1680     CPX #1
1690     BNE CHNYB
1700 * SINCE THE COLOR ITSELF IS IN
1710 * THE HI-NYBBLE OF THE GIVEN
1720 * COLOR, MOVE IT TO THE LOWER
1730 * NYBBLE SINCE THE X-POS IS ODD
1740     LSR A
1750     LSR A
1760     LSR A
1770     LSR A
1780     JMP POKCOL
1790 CHNYB
1800     AND #$F0       ;SINCE XPOS EVEN
1810 POKCOL
1820     ORA T8
1830 * PUT IT INTO THE SCREEN
1840     STA (YBYT),Y
1850     RTS
1860 * BIT MASKS TABLE
1870 BITABL .BYTE $0F,$F0
1880 *****
1890 * BASIC ENTRY:
1900 * Q=USR(BASDRW2,X,Y,COL)
1910 *
1920 * THIS SUBR DRAWS A LINE FROM
1930 * THE CURRENT CURSOR POSITION
1940 * TO THE GIVEN X,Y USING

```

```

1950 * THE GIVEN COLOR
1960 *
1970 * FIRST MAKE SURE THERE ARE 3
1980 * (NO MORE, NO LESS) ARGUMENTS
1990 *
2000 BASDRW2
2010     PLA
2020     CMP #3
2030     BEQ GETARGS
2040     JMP TRAP      ;WRONG # OF ARGS
2050 GETARGS
2060     PLA
2070     PLA
2080     TAX          ;DEST-X
2090     PLA
2100     PLA
2110     TAY          ;DEST-Y
2120     LDA COLCRS  ;CURSOR X-POS
2130     STA FX       ;FROM-X
2140     LDA ROWCRS  ;CURSOR Y-POS
2150     STA FY       ;FROM-Y
2160     PLA
2170     PLA
2180 * CHECK FOR ILLEGAL DRAWTO
2190     JSR BOUNDCK
2200 * ML ENTRY: A=COLOR
2210 *           X=DEST-X
2220 *           Y=DEST-Y
2230 DRT256
2240     STX TMP1     ;SAVE X
2250     STY YD       ;SAVE Y
2260 * PLOT THE DESTINATION PIXEL
2270     JSR PLT256
2280     LDY YD       ;RESTORE Y
2290 * LOOP TO SET XD,YD,XACC,YACC,
2300 * DELTX,DELT Y TO 0
2310     LDX #6
2320     LDA #0
2330 POKZ0
2340     STA XD-1,X
2350     DEX
2360     BNE POKZ0
2370 * THE FOLLOWING ROUTINE IS BASED
2380 * ON THE VECTOR ROUTINE PRINTED
2390 * IN ANALOG, ISSUE #18, IN THE
2400 * "BASIC TRAINING" COLUMN BY
2410 * TOM HUDSON
2420 * INIT THE X VARIABLES
2430     LDX TMP1
2440     CPX FX
2450     BEQ SGNY
2460     BCC NEGXD
2470     TXA          ;DELT X=TX-FX
2480     SBC FX       ;CARRY WAS SET
2490     INC XD       ;XD=1
2500     BNE SAVDLX  ;ALWAYS
2510 NEGXD
2520     DEC XD       ;XD=-1 ($FF)
2530     STX TMP1

```

```

2540     LDA FX           ;DELTX=FX-TX
2550     SEC
2560     SBC TMP1
2570 SAVDLX
2580     STA DELTX
2590 * INIT THE Y VARIABLES
2600 SGNY
2610     CPY FY
2620     BEQ INTCNT
2630     BCC NEGY
2640     TYA             ;DELTY=YD-FY
2650     SBC FY         ;CARRY WAS SET
2660     INC YD         ;YD=1
2670     BNE SAVDLY    ;ALWAYS
2680 NEGY
2690     DEC YD         ;YD=-1 ($FF)
2700     STY TMP1
2710     LDA FY         ;DELTY=FY-TY
2720     SEC
2730     SBC TMP1
2740 SAVDLY
2750     STA DELTY
2760 INTCNT
2770     LDA DELTX
2780     CMP DELTY
2790     BCC XLTY
2800     STA YACC
2810     BCS INTEPT   ;ALWAYS
2820 * (X LESS THAN Y)
2830 XLTY
2840     LDA DELTY
2850     STA XACC
2860 INTEPT
2870     STA COUNT
2880     STA EPOINT
2890     LSR XACC      ;XACC=XACC/2
2900     LSR YACC      ;YACC=YACC/2
2910 * EXIT IF DESTINATION X,Y
2920 * SAME AS CURRENT X,Y
2930     LDA COUNT
2940     BEQ EXDRT
2950 *
2960 * THE MAIN LOOP!
2970 XCALC
2980     LDA XACC      ;CHANGE X
2990     CLC
3000     ADC DELTX
3010     STA XACC
3020     CMP EPOINT
3030     BCC YCALC
3040     SBC EPOINT   ;CARRY WAS SET
3050     STA XACC
3060     LDA FX
3070     CLC
3080     ADC XD
3090     STA FX
3100 YCALC
3110     LDA YACC      ;CHANGE Y
3120     CLC

```

```

3130     ADC DELTY
3140     STA YACC
3150     CMP EPOINT
3160     BCC PLIT
3170     SBC EPOINT ;CARRY WAS SET
3180     STA YACC
3190     LDA FY
3200     CLC
3210     ADC YD
3220     STA FY
3230 * PLOT THE CALCULATED POINT
3240 PLIT
3250     LDX FX
3260     LDY FY
3270     JSR PSAMCOL ;PLOT SAME COLOR
3280     DEC COUNT
3290     BNE XCALC ;DO MORE POINTS
3300 EXDRT
3310     RTS
3320 *****
3330 * BASIC ENTRY:
3340 * Q=USR(BASEXGT)
3350 *
3360 * THIS SUBR RESTORES THE SCREEN
3370 * TO GRAPHICS 0 AND RESETS OS
3380 * INTERRUPT POINTERS, ETC.
3390 * IT STILL PROTECTS APAC
3400 *
3410 BASEXGT
3420     PLA
3430     BEQ EXITGT ;NO PARAMETERS!
3440     JMP TRAP
3450 * ML ENTRY:
3460 * (REGISTERS IGNORED)
3470 EXITGT
3480     LDA VCOUNT ;MAKE SURE
3490     CMP #64 ;WE AREN'T NEAR
3500     BCS EXITGT ;A VBI
3510     LDA SAVVBI ;RESTORE OS VBI
3520     STA VVBLKI
3530     LDA SAVVBI+1
3540     STA VVBLKI+1
3550 *
3560     LDA #64
3570     STA NMIEN ;NO DLI'S
3580     LDX #$00 ;IOCB 0
3590     LDA #$0C ;CLOSE
3600     STA ICCOM
3610     JSR CIOV ;GO DO IT
3620 *
3630     LDX #$00 ;IOCB 0
3640     STX ICAUX1+1
3650     LDA #$03 ;OPEN
3660     STA ICCOM
3670 * POINT TO "E:", BELOW
3680     LDA # <SCRDEV
3690     STA ICBADR
3700     LDA # >SCRDEV
3710     STA ICBADR+1

```

```

3720     LDA #0C      ;READ/WRITE
3730     STA ICAUX1
3740     JSR CIOV     ;GO DO IT
3750 *
3760     LDA #7F
3770     STA NMIEN    ;NORML INTRUPTS
3780 * PROTECT APAC BY MOVING THE
3790 * BASIC AND OS POINTERS UP
3800 SETMEM
3810     LDA # <ENDAPAC ;LO-BYTE
3820     STA MEMLO
3830     STA LOMEM
3840     LDA # >ENDAPAC ;HI-BYTE
3850     STA MEMLO+1
3860     STA LOMEM+1
3870     RTS
3880 SAVVBI .BYTE 0,0
3890 SCRDEV .BYTE "E:"
3900 *****
3910 * BASIC ENTRY:
3920 * Q=USR(BASINGT)
3930 *
3940 * THIS SUBR SETS UP THE SPECIAL
3950 * APAC SCREEN WITH ALTERNATING
3960 * LINES OF LUMINANCE AND COLOR
3970 *
3980 BASINGT
3990     PLA
4000     BEQ INITGT   ;ZERO ARGS
4010     JMP TRAP
4020 * ML ENTRY:
4030 * (REGISTERS IGNORED)
4040 INITGT
4050     LDA #0
4060     STA SDMCTL   ;DISABLE SCR
4070     STA NMIEN    ;NO INTERRUPTS
4080     STA ROWCRS  ;CURSOR=(0,0)
4090     STA COLCRS
4100     STA COLCRS+1
4110     STA COLOR4  ;BAKGRND=BLACK
4120     STA COLSAV  ;DEFAULT COL=0
4130     STA YBYT    ;MY POINTR TO DL
4140     STA SDLSTL  ;OS POINTR TO DL
4150     TAY        ;Y=0
4160     LDA RAMTOP  ;HI-BYTE OF
4170     SEC        ;DLIST ADR=
4180     SBC #32    ; RAMTOP-32
4190     STA YBYT+1
4200     STA SDLSTL+1
4210 * CREATE THE DISPLAY LIST
4220 *
4230 * THE APAC DISPLAY LIST IS MUCH
4240 * THE SAME AS A GRAPHICS 8 DL
4250 * EXCEPT THERE ARE DLI'S ON
4260 * EVERY LINE, INCL THE LAST
4270 * BLANK SCAN LINE AT TOP OF THE
4280 * SCREEN
4290     LDA #112    ;CMD, 8 BLNK LNS
4300 NXTBNK

```



```

4310     STA (YBYT),Y
4320     INY
4330     CPY #3
4340     BNE NXTBNK
4350     LDA #\$90      ;8 BLNKS+DLI
4360     STA (YBYT),Y
4370     LDA #\$CF      ;MODE 15+DLI+LMS
4380     INY           ;Y=4
4390     STA (YBYT),Y
4400     LDA #0        ;LO-BYTE,  SCRN
4410     INY           ;Y=5
4420     STA (YBYT),Y
4430     STA SCREEN
4440     LDX YBYT+1   ;HI-BYTE OF  SCRN
4450     INX          ;IS RAMTOP-31
4460     TXA
4470     STA SCREEN+1
4480     INY           ;Y=6
4490     STA (YBYT),Y
4500     INY           ;Y=7
4510     LDA #\$8F    ;ANTIC 15+DLI
4520 NXANTLN
4530     STA (YBYT),Y
4540     INY
4550     CPY #200
4560     BNE NXANTLN
4570     LDY #102     ;102TH BYTE,  DL
4580     LDA #\$CF    ;MODE 15+DLI+LMS
4590     STA (YBYT),Y
4600     LDA #0      ;ADR OF 2ND HALF
4610     INY         ;OF SCRN IS:
4620     STA (YBYT),Y
4630     LDA RAMTOP  ;(RAMTOP-16)*256
4640     SEC
4650     SBC #16
4660     INY         ;Y=104
4670     STA (YBYT),Y
4680     LDY #200
4690     LDA #65
4700     STA (YBYT),Y
4710     INY         ;Y=201
4720     LDA #0      ;LO-BYTE,  ADR DL
4730     STA (YBYT),Y
4740     INY         ;Y=202
4750     LDA YBYT+1 ;HI-BYTE,  ADR DL
4760     STA (YBYT),Y
4770 * CLEAR OUT SCREEN MEMORY
4780 CLRSCR
4790     LDA SCREEN
4800     STA ZTEMP
4810     LDA SCREEN+1
4820     STA ZTEMP+1
4830     CLC
4840     ADC #30     ;LAST PAGE OF
4850     STA TMP1   ;MEM TO CLEAR+1
4860 SPAGE
4870     LDY #0
4880     TYA       ;A=0
4890 ZERSB

```

```

4900     STA (ZTEMP),Y
4910     DEY
4920     BNE ZERSB
4930     INC ZTEMP+1
4940     LDA ZTEMP+1
4950     CMP TMP1      ;END?
4960     BCC SPAGE    ;NOPE
4970 * INITIALIZE THE TABLE
4980 * OF ADDRESSES THAT POINT TO THE
4990 * 96 APAC SCREEN LINES
5000     LDA SCREEN+1
5010     STA ZTEMP+1 ;HI-BYTE
5020     LDY #$00
5030 NXTLN80
5040     LDA ZTEMP    ;SAVE LO
5050     STA SCRALO,Y
5060     LDA ZTEMP+1 ;SAVE HI
5070     STA SCRAHI,Y
5080     CLC
5090     LDA ZTEMP
5100     ADC #80      ;80 BYTES PER
5110     STA ZTEMP    ;APAC LINE
5120     BCC NXTY
5130     INC ZTEMP+1
5140 NXTY
5150     INY
5160     CPY #96      ;96 LINES TO DO
5170     BNE NXTLN80
5180     LDA VVBLKI   ;SAVE THE OS VBI
5190     STA SAVVBI
5200     LDA VVBLKI+1
5210     STA SAVVBI+1
5220 * POINT TO THE APAC IMM VBI
5230     LDA #6        ;STAGE 1 VBI
5240     LDX # >IVBI  ;HI-BYTE
5250     LDY # <IVBI  ;LO-BYTE
5260     JSR SETVBV   ;SET IT
5270 * POINT TO THE FIRST DLI
5280     LDA # <DLI1
5290     STA VDLSLST
5300     LDA # >DLI1
5310     STA VDLSLST+1
5320     LDA #$40     ;GRAPHICS 9
5330     STA GPRIOR
5340     LDA #$C0     ;ALL INTERRUPTS
5350     STA NMIEN
5360     LDA #34
5370     STA SDMCTL   ;SCREEN ON
5380 LEAVE
5390     RTS
5400 *****
5410 * THE RESET HANDLER:
5420 * THE JSR $FFFF WILL POINT TO
5430 * DOS AFTER LOADING APAC
5440 * (SEE SYSSET, BELOW)
5450 *
5460 WRMSTRT
5470     JSR $FFFF
5480 * POINT THE OS RESET VECTORS

```

```

5490 * TO WRMSTRT
5500 SETVEC
5510     LDA # <WRMSTRT ;LO-BYTE
5520     STA DOSINI
5530     STA CASINI
5540     LDA # >WRMSTRT ;HI-BYTE
5550     STA DOSINI+1
5560     STA CASINI+1
5570 * GO SET THE LOMEM POINTERS
5580     JMP SETMEM
5590 *****
5600 * MAKE SURE X<80 AND Y<96
5610 *
5620 BOUNDCK
5630     CPX #80
5640     BCC CKYPOS
5650     BCS ERBND ;X>79, EXIT APAC
5660 CKYPOS
5670     CPY #96
5680     BCC LEAVE ;ALL'S WELL
5690 ERBND
5700     PLA ;Y>95, SO PULL
5710     PLA ;JSR TO BOUNDCK
5720     RTS ;EXIT APAC
5730 *****
5740 * PULL OFF EXCESS ARGUMENTS
5750 *
5760 TRAP
5770     TAX
5780     BEQ EXTRAP ;NO ARGS TO PULL
5790 DELARG
5800     PLA ;PULL HI
5810     PLA ;PULL LO
5820     DEX
5830     BNE DELARG ;DO MORE ARGS
5840 EXTRAP
5850     RTS
5860 *****
5870 * APAC'S IMM MODE VBI KEEPS
5880 * EVERYTHING TIMED RIGHT
5890 *
5900 IVBI
5910     PHA ;SAVE ACC.
5920     LDA # $C0 ;ENABLE
5930     STA NMIEN ;ALL INTERRUPTS
5940     LDA # <DLI1 ;POINT TO
5950     STA VDLSLST ;FIRST DLI
5960     STA COLOR4 ;BACKGRND=BLACK
5970     PLA ;RESTORE ACC.
5980     JMP SYSVBV ;DO OS IMM VBI
5990 *****
6000 * THE DLI'S TOGGLE- I.E. THEY
6010 * APPEAR EVERY OTHER LINE,
6020 * ALTERNATING GRAPHICS 9 & 11
6030 *
6040 DLI1
6050     PHA ;SAVE ACC.
6060     LDA # <DLI2
6070     STA VDLSLST ;POINT TO DLI2

```

```

6080     LDA #$40
6090     STA WSYNC
6100     STA PRIOR      ;GRAPHICS 9
6110     PLA           ;RESTORE ACC.
6120     RTI
6130 DLI2
6140     PHA           ;SAVE ACC.
6150     LDA # <DLI1
6160     STA VDSLST   ;POINT TO DLI1
6170     LDA #$C0
6180     STA WSYNC
6190     STA PRIOR      ;GRAPHICS 11
6200     PLA           ;RESTORE ACC.
6210     RTI
6220 *****
6230 SCRALO *= *+96
6240 SCRAHI *= *+96
6250 ENDAPAC = *
6260 *****
6270 * SYSSET INITIALIZES WRMSTRT
6280 * (POINTS A JSR THERE TO DOS)
6290 * ALSO JSR'S TO SETVEC
6300 * WHICH INIT'S APAC'S RESET TRAP
6310 * AND SELF-PROTECTION
6320 *
6330     *= $4000
6340 SYSSET
6350     LDA DOSINI   ;MODIFY CODE
6360     STA WRMSTRT+1 ; IN SUBR
6370     LDA DOSINI+1 ; WARMSTRT
6380     STA WRMSTRT+2
6390     JSR SETVEC
6400     LDA TRAMSZ   ;IF 1, CART IN
6410     BNE GOCART
6420     RTS           ;(NO CARTRIDGE)
6430 GOCART
6440     LDA #0
6450     STA WARMST   ;DO A WARMSTART
6460     JMP $A000    ;TO CARTRIDGE

```

```

10 *****
20 * APAC KAL DEMO, V2.0      *
30 * CREATED BY THOMAS TANIDA *
40 * FIRST STARTED: 1/31/87  *
50 * LAST REVISED: 2/7/87   *
60 *****
70     *= $6000
80     .OPT NO EJECT
90     .TAB 8,12,30
0100 *****
0110 ZTEMP = $80
0120 DLE = $82
0130 COLOR = $84
0140 COORDS = $85
0150 DLTX = $89
0160 OFFX = $8D
0170 PNTPOS = $96
0180 TEMP = $A0
0190 STORE = $B0

```

```

0200 *
0210 * SYSTEM EQUATES
0220 *
0230 KEYDEL = $02F1 ;753
0240 KEYPRS = $02FC ;754
0250 CONSOL = $D01F ;53279
0260 RANDOM = $D20A ;53770
0270 *
0280 * APAC SYS EQUATES
0290 *
0300 APACPLOT = $230C
0310 APACDRAW = $230F
0320 APACEXIT = $2312
0330 APACINIT = $2315
0340 *
0350 *****
0360 STARTUP
0370     JSR APACINIT ;SET UP APAC
0380     LDX #39      ;INIT VARS
0390     STX OFFX     ;OFFSETS
0400     STX OFFX+1
0410     INX
0420     STX OFFX+3
0430 *
0440     LDA #$00     ;ZERO OUT
0450     LDX #$08     ;$A5-$AC
0460 INTO
0470     STA COORDS-1,X
0480     DEX
0490     BNE INTO
0500     STA COLOR    ;1ST COLOR=BLACK
0510 LOOP
0520     LDA #$FF     ;CLEAR KEYBD REG
0530     STA KEYPRS
0540     JSR POSSET   ;LINE 1
0550     JSR POSSET   ;LINE 2
0560     LDA RANDOM   ;GET A RAND #
0570     AND #$1F     ;FROM 0-31
0580     BNE GOMOV    ;31 IN 32
0590     JSR DIRSET   ;1 IN 32
0600 GOMOV
0610     JSR MAKMOV
0620     INC COLOR    ;CHANGE COLOR
0630     LDA CONSOL   ;GET CONSOLE KEY
0640     CMP #6       ;START PRESSED?
0650     BEQ STARTUP  ;YEP- CLEAR SCR
0660     LDA KEYPRS   ;GET KEY
0670     CMP #33      ;SPACE PRESSED?
0680     BEQ PAUSE    ;YEP- PAUSE
0690     CMP #28      ;ESC PRESSED?
0700     BNE LOOP     ;NO- MAIN LOOP
0710     JMP APACEXIT
0720 *****
0730 PAUSE
0740     LDA KEYDEL   ;WAIT FOR KEY
0750     BNE PAUSE    ;TO BE RELEASED
0760     LDA #$FF     ;CLEAR KEYBD
0770     STA KEYPRS
0780 GETSPC

```

```

0790     LDA KEYPRS   ;GET KEYCODE
0800     CMP #33      ;SPACE?
0810     BNE GETSPC   ;WAIT FOR SPACE
0820     JMP LOOP     ;BACK TO MAIN
0830 *****
0840 POSSET
0850     LDX #2
0860 NXP2
0870     LDA COORDS+1,X ;ROTATE X2,Y2
0880     JSR SUB256
0890     STA COORDS+1,X
0900     DEX
0910     BNE NXP2
0920     JSR PNTSCR
0930 *
0940     LDX #2
0950 NXP1
0960     LDA COORDS-1,X ;ROTATE X1,Y1
0970     JSR SUB256
0980     STA COORDS-1,X
0990     DEX
1000     BNE NXP1
1010     JMP PNTSCR
1020 *****
1030 SUB256
1040     EOR #$FF     ;ACC=256-ACC
1050     CLC
1060     ADC #1
1070     RTS
1080 *****
1090 * DIRSET CHANGES THE DIRECTION
1100 * OF THE MOVEMENT OF LINES YOU
1110 * SEE. OTHERWISE, THE KAL WOULD
1120 * BE REPETITIVE AND BORING.
1130 *
1140 DIRSET
1150     LDX #4
1160 GETPLM
1170     LDA RANDOM   ;GET A RANDOM #
1180     AND #$03     ;FROM 0-3
1190     BEQ GETPLM   ;BUT NOT 0
1200     SEC          ;MAKE IT -1,0,1
1210     SBC #2
1220     STA DLTx-1,X ;SAVE IT
1230     DEX          ;4 TIMES
1240     BNE GETPLM
1250     RTS
1260 *****
1270 MAKMOV
1280     LDX #4
1290 NXTPSX
1300     LDA COORDS-1,X ;GET COORD
1310     CLC
1320     ADC DLTx-1,X ;ADD -1,0 OR 1
1330 * ESSENTIALLY THESE NEXT FEW
1340 * LINES DO A "ACC MOD 40"
1350 * BY SUBTRACTING 40 EACH TIME
1360 * ACC>=40 UNTIL ACC<40
1370 *

```

```

1380 * THE PURPOSE IS TO MAKE SURE
1390 * THE COORDS DON'T GO OUTSIDE
1400 * SCREEN LIMITS (THERE ARE 4
1410 * QUADRANTS 40 BY 40)
1420 MOD40
1430     CMP #40
1440     BCC LT40     ;LESS THAN 40
1450     SEC
1460     SBC #40     ;A=A-40
1470     BCS MOD40   ;GO BACK
1480 LT40
1490     STA COORDS-1,X ;SAVE IT BACK
1500     DEX
1510     BNE NXTPSX   ;NEXT POSITION
1520     RTS
1530 *****
1540 PNTSCR
1550     LDY #4
1560 NXPERM
1570     LDX OFOFF-1,Y ;ALTERNATE
1580     LDA COORDS-1,Y ;THE COORDS
1590     CLC           ;PLOTTED
1600     ADC OFFX,X   ;TO REFLECT
1610     LDX PNTOFF-1,Y ;IN EACH
1620     STA PNTPOS,X ;PART OF THE
1630     DEY           ;SCREEN
1640     BNE NXPERM
1650     JSR PUTONSCR ;DRAW ON SCR
1660 *
1670     LDY #4       ;SAME AS ABOVE
1680 NXCOMB
1690     LDX OFOFF+1,Y ;SEE END OF
1700     LDA COORDS-1,Y ;KAL FOR THE
1710     CLC           ;REARRANGEMENT
1720     ADC OFFX,X   ;OF THE COORDS
1730     LDX PNTOFF+1,Y
1740     STA PNTPOS,X
1750     DEY
1760     BNE NXCOMB
1770 * (FALLS THROUGH TO BELOW)
1780 *****
1790 PUTONSCR
1800     LDA COLOR    ;GET COLOR
1810     LDX PNTPOS   ;X-POS
1820     LDY PNTPOS+1 ;Y-POS
1830     JSR APACPLOT ;PLOT X1,Y1
1840     LDA COLOR    ;GET COLOR
1850     LDX PNTPOS+2 ;DEST X-POS
1860     LDY PNTPOS+3 ;DEST Y-POS
1870     JSR APACDRAW ;DRAWTO X2,Y2
1880 *
1890     LDA COLOR    ;GET COLOR
1900     LDX PNTPOS+1 ;X-POS
1910     LDY PNTPOS+2 ;Y-POS
1920     JSR APACPLOT ;PLOT Y1,X2
1930     LDA COLOR    ;GET COLOR
1940     LDX PNTPOS   ;DEST X-POS
1950     LDY PNTPOS+3 ;DEST Y-POS
1960     JMP APACDRAW ;DRAWTO X1,Y2

```

1970 *****

1980 OFOFF .BYTE 0,0,1,1,0,0

1990 PNTOFF .BYTE 0,2,1,3,0,2