```
LOCATION OBJECT CODE LINE     SOURCE LINE

    1E45 D9          6973                 EXX
    1E46 D1          6974                 POP            DE
    1E47 2A8006      6975                 LD             HL,[WORK_BUFFER]
    1E4A 010008      6976                 LD             BC,8
    1E4D 09          6977                 ADD            HL,BC
    1E4E FD210004    6978                 LD             IY,4
    1E52 CD1C27      6979                 CALL           PUT_VRAM_
                     6980
                     6981 *   END_IF
    1E55             6982 END_IF_4_GRAPHICS
                     6983
                     6984 *   DESTINATION := DESTINATION + 4
    1E55 D9          6985                 EXX
    1E56 23          6986                 INC            HL
    1E57 23          6987                 INC            HL
    1E58 23          6988                 INC            HL
    1E59 23          6989                 INC            HL
                     6990
                     6991 * END
    1E5A C31D8C      6992                 JP             RETURN_HERE
                     6993
                     6994
                     6995
                     6996
    1E5D             6997 COLOR_TEST                     ; TESTS WHETHER PATTERN GENERATORS ARE
                     6998                                ; BEING MANIPULATED AND WHETHER THE
                     6999                                ; GRAPHICS MODE IS 2. IF SO THE ABOVE
                     7000                                ; ROUTINES NEED TO DEAL WITH THE COLOR
                     7001                                ; GENERATORS THAT CORRESPOND TO THE
                     7002                                ; PATTERN GENERATORS THEY ARE OPERATING
                     7003                                ; ON.
                     7004
                     7005                                ; NOT INPUTS, RETURNS WITH TRUE (1) IN
                     7006                                ; A IF CONDITION IS TRUE, FALSE (0) IF
                     7007                                ; NOT.
                     7008
                     7009
                     7010 * BEGIN COLOR_TEST
                     7011
                     7012 * CHECK TABLE CODE IN A'
    1E5D 08          7013                 EX             AF,AF'
    1E5E F5          7014                 PUSH           AF
    1E5F 08          7015                 EX             AF,AF'
    1E60 F1          7016                 POP            AF
    1E61 FE03        7017                 CP             PATTERN_GEN
    1E63 200A        7018                 JR             NZ,EXIT_FALSE
                     7019
                     7020 * CHECK MODE
    1E65 2173C3      7021                 LD             HL,VDP_MODE_WORD
    1E68 CB4E        7022                 BIT            1,[HL]
    1E6A 2803        7023                 JR             Z,EXIT_FALSE
                     7024
                     7025 * EXIT HERE IF TRUE
    1E6C 3E01        7026                 LD             A,TRUE
    1E6E C9          7027                 RET
                     7028
                     7029 * EXIT HERE IF FALSE
```

```
CATION OBJECT CODE LINE     SOURCE LINE

  1E6F 3E00        7030 EXIT_FALSE      LD              A,FALSE
  1E71 C9          7031                 RET
                   7032
                   7033
                   7034
  1E72             7035 PUT_TABLE
                   7036                                 ; PUTS THE CONTENTS OF WORK_BUFFER[0..15]
                   7037                                 ; IN VRAM AT THE GIVEN DESTINATION.
                   7038
  1E72 08          7039                 EX              AF,AF'
  1E73 F5          7040                 PUSH            AF
  1E74 08          7041                 EX              AF,AF'
  1E75 F1          7042                 POP             AF
  1E76 D9          7043                 EXX
  1E77 E5          7044                 PUSH            HL
  1E78 D9          7045                 EXX
  1E79 D1          7046                 POP             DE
  1E7A 2A8006      7047                 LD              HL,[WORK_BUFFER]
  1E7D 010008      7048                 LD              BC,8
  1E80 09          7049                 ADD             HL,BC
  1E81 FD210001    7050                 LD              IY,1
  1E85 CD1C27      7051                 CALL            PUT_VRAM_
  1E88 C9          7052                 RET
                   7053
                   7054
  1E89             7055 GET_COLOR
                   7056                                 ; GETS THE COLOR INFORMATION FROM
                   7057                                 ; THE APPROPRIATE PLACE IN VRAM
                   7058
  1E89 3E04        7059                 LD              A,COLOR_TABLE
  1E8B D9          7060                 EXX
  1E8C D5          7061                 PUSH            DE
  1E8D D9          7062                 EXX
  1E8E D1          7063                 POP             DE
  1E8F 2A8006      7064                 LD              HL,[WORK_BUFFER]
  1E92 FD210001    7065                 LD              IY,1
  1E96 CD1BA3      7066                 CALL            GET_VRAM_
  1E99 C9          7067                 RET
                   7068
                   7069
  1E9A             7070 PUT_COLOR
                   7071                                 ; PUTS COLOR INFORMATION IN THE
                   7072                                 ; APPROPRIATE PLACE IN  VRAM
                   7073
  1E9A 3E04        7074                 LD              A,COLOR_TABLE
  1E9C D9          7075                 EXX
  1E9D E5          7076                 PUSH            HL
  1E9E D9          7077                 EXX
  1E9F D1          7078                 POP             DE
  1EA0 2A8006      7079                 LD              HL,[WORK_BUFFER]
  1EA3 FD210001    7080                 LD              IY,1
  1EA7 CD1C27      7081                 CALL            PUT_VRAM_
  1EAA C9          7082                 RET
                   7083  PROG
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     7085
                     7086 * THE ROUTINES IN THIS MODULE TAKE A SINGLE 8-BYTE BLOCK AS INPUT AND
                     7087 * PRODUCE 4 8-BYTE BLOCKS AS OUTPUT. THEY PERFORM A 2-TO-1 EXPANSION
                     7088 * AND A SIMPLE QUADRUPLING OPEATION RESPECTIVELY.
                     7089
                     7090
                     7091
                     7092                 GLB             MAGNIFY
                     7093                 GLB             QUADRUPLE
                     7094 * NAMES OF ENTRY POINTS
                     7095
                     7096
                     7097
                     7098
   1EAB              7099 MAGNIFY                         ; PERFORM A 2-TO-1 EXPANSION ON AN
                     7100                                 ; 8-BYTE BLOCK OF DATA.
                     7101
                     7102                                 ; SOURCE POINTER IN HL, DESTINATION
                     7103                                 ; POINTER IN DE.
                     7104
                     7105                                 ; DESTROYS IX,IY,AF,BC,DE,HL
                     7106
          <000A>     7107 BYTE_COUNT      EQU             BC
          <000E>     7108 SOURCE          EQU             IX
          <000F>     7109 DESTINATION     EQU             IY
                     7110 * STANDARD NAMES FOR REGISTERS IN THIS ROUTINE
                     7111
                     7112 * BEGIN
   1EAB E5           7113                 PUSH            HL              ; SET UP POINTERS
   1EAC DDE1         7114                 POP             SOURCE
   1EAE D5           7115                 PUSH            DE
   1EAF FDE1         7116                 POP             DESTINATION
                     7117
                     7118 *   BYTE_COUNT := 8
   1EB1 010008       7119                 LD              BYTE_COUNT,8
                     7120
                     7121 *   REPEAT
   1EB4              7122 MAG_LOOP
                     7123
                     7124 *     EXPAND A BYTE FROM SOURCE
   1EB4 DD7E00       7125                 LD              A,(SOURCE+0)
   1EB7 DD23         7126                 INC             SOURCE
   1EB9 57           7127                 LD              D,A
   1EBA 1E04         7128                 LD              E,4
   1EBC CB17         7129 EXP_1           RL              A               ; DOUBLE BITS IN HIGH
   1EBE CB14         7130                 RL              H               ; ORDER NIBBLE
   1EC0 CB12         7131                 RL              D
   1EC2 CB14         7132                 RL              H
   1EC4 1D           7133                 DEC             E
   1EC5 20F5         7134                 JR              NZ,EXP_1
   1EC7 1E04         7135                 LD              E,4
   1EC9 CB17         7136 EXP_2           RL              A               ; DOUBLE BITS IN LOW
   1ECB CB15         7137                 RL              L               ; ORDER NIBBLE
   1ECD CB12         7138                 RL              D
   1ECF CB15         7139                 RL              L
   1ED1 1D           7140                 DEC             E
   1ED2 20F5         7141                 JR              NZ,EXP_2
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     7142
                     7143 *     WRITE IT TO DESTINATION
    1ED4 FD7400      7144                LD              [DESTINATION+0],H
    1ED7 FD7510      7145                LD              [DESTINATION+16],L
    1EDA FD23        7146                INC             DESTINATION
    1EDC FD7400      7147                LD              [DESTINATION+0],H
    1EDF FD7510      7148                LD              [DESTINATION+16],L
    1EE2 FD23        7149                INC             DESTINATION
                     7150
                     7151 *     DECREMENT BYTE_COUNT
    1EE4 08          7152                DEC             BYTE_COUNT
                     7153
                     7154 *   UNTIL BYTE_COUNT = 0
    1EE5 79          7155                LD              A,C
    1EE6 B0          7156                OR              B
    1EE7 20CB        7157                JR              NZ,MAG_LOOP
                     7158
                     7159 * END
    1EE9 C9          7160                RET
                     7161
    1EEA             7162 QUADRUPLE                       ; PERFORM A QUADRUPLING ON AN
                     7163                                 ; 8-BYTE BLOCK OF DATA.
                     7164
                     7165                                 ; SOURCE POINTER IN HL, DESTINATION
                     7166                                 ; POINTER IN DE.
                     7167
                     7168                                 ; DESTROYS AF,BC,DE,HL,IY
                     7169
                     7170 * BEGIN
                     7171
                     7172 *   BYTE_COUNT := 16
    1EEA 010010      7173                LD              BYTE_COUNT,16
                     7174
                     7175 *   SAVE SOURCE
    1EED E5          7176                PUSH            HL
                     7177
                     7178 *   REPEAT
    1EEE             7179 QUAD_LOOP
                     7180
                     7181 *     GET A BYTE FROM SOURCE
    1EEE 7E          7182                LD              A,[HL]
    1EEF 23          7183                INC             HL
                     7184
                     7185 *     WRITE IT TWICE TO DESTINATION
    1EF0 12          7186                LD              [DE],A
    1EF1 13          7187                INC             DE
    1EF2 12          7188                LD              [DE],A
    1EF3 13          7189                INC             DE
                     7190
                     7191 *     DECREMENT BYTE_COUNT
    1EF4 0B          7192                DEC             BYTE_COUNT
                     7193
                     7194 *      IF BYTE_COUNT = 8 THEN RESTORE SOURCE
    1EF5 79          7195                LD              A,C
    1EF6 FE08        7196                CP              8
    1EF8 2001        7197                JR              NZ,SKIPZZ
    1EFA E1          7198                POP             HL
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    1EFB             7199 SKIPZZ
                     7200
                     7201 *   UNTIL BYTE_COUNT = 0
    1EFB 79          7202                 LD              A,C
    1EFC B0          7203                 OR              B
    1EFD 20EF        7204                 JR              NZ,QUAD_LOOP
                     7205
                     7206 * END
    1EFF C9          7207                 RET
                     7208  PROG
```

```
CATION OBJECT CODE LINE     SOURCE LINE

                  7210
                  7211 * THE ROUTINES IN THIS FILE TAKE A SINGLE 8-BYTE BLOCK AS INPUT
                  7212 * AND OPERATE ON IT PRODUCING A SINGLE 8-BYTE BLOCK AS OUTPUT.
                  7213 * THEY PERFORM MIRRORING AROUND THE VERTICAL AXIS, MIRRORING
                  7214 * AROUND THE HORIZONTAL AXIS, AND 90 DEGREE ROTATION.
                  7215
                  7216                GLB         MIRROR_L_R
                  7217                GLB         ROTATE
                  7218                GLB         MIRROR_U_D
                  7219
                  7220
  1F00            7221 MIRROR_L_R                 ; REFLECTS AN 8X8 PIXEL DATA BLOCK
                  7222                            ; AROUND THE VERTICAL AXIS.
                  7223
                  7224                            ; SOURCE IN HL, DEST IN DE
                  7225                            ; DESTROYS AF,BC,DE,HL
                  7226
  1F00 010008     7227          LD    BC,8        ;SET BLOCK BYTE COUNT
  1F03            7228 MIR_L_R10
  1F03 46         7229          LD    B,(HL)      ;GET SOURCE BYTE
  1F04 3E80       7230          LD    A,80H       ;SET WORK REGISTER MASK
  1F06            7231 MIR_L_R20
  1F06 CB10       7232          RL    B           ;PUT SOURCE IN CARRY
  1F08 1F         7233          RRA               ;PUT CARRY INTO WORK REGISTER
  1F09 30FB       7234          JR    NC,MIR_L_R20 ;CONTINUE UNTIL MASK BIT IS IN CARRY
  1F0B 12         7235          LD    (DE),A      ;WRITE MIRRORED BYTE TO DESTINATION
  1F0C 23         7236          INC   HL          ;ADVANCE POINTERS
  1F0D 13         7237          INC   DE
  1F0E 0D         7238          DEC   C
  1F0F 20F2       7239          JR    NZ,MIR_L_R10 ;CHECK COUNT
  1F11            7240 MIR_L_RX
  1F11 C9         7241          RET
                  7242
                  7243
                  7244
  1F12            7245 ROTATE                     ; ROTATE OBJECT 90 DEGREES
                  7246
                  7247                            ; SOURCE IN HL DESTINATION IN DE.
                  7248                            ; DESTROYS AF,BC,DE,HL
                  7249
  1F12 E5         7250          PUSH  HL
  1F13 DDE1       7251          POP   IX
  1F15 EB         7252          EX    DE,HL
  1F16 010008     7253          LD    BC,8
  1F19            7254 TRANSP_10
  1F19 DDCB0016   7255          RL    (IX+0)      ;PUT HI BIT OF FIRST SOURCE BYTE IN CARRY
  1F1D CB1E       7256          RR    (HL)        ;PUT CARRY IN DESTINATION BYTE
  1F1F DDCB0116   7257          RL    (IX+1)
  1F23 CB1E       7258          RR    (HL)
  1F25 DDCB0216   7259          RL    (IX+2)
  1F29 CB1E       7260          RR    (HL)
  1F2B DDCB0316   7261          RL    (IX+3)
  1F2F CB1E       7262          RR    (HL)
  1F31 DDCB0416   7263          RL    (IX+4)
  1F35 CB1E       7264          RR    (HL)
  1F37 DDCB0516   7265          RL    (IX+5)
  1F3B CB1E       7266          RR    (HL)
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    1F3D DDCB0616    7267            RL    (IX+6)
    1F41 CB1E        7268            RR    (HL)
    1F43 DDCB0716    7269            RL    (IX+7)
    1F47 CB1E        7270            RR    (HL)
    1F49 23          7271            INC   HL
    1F4A 0D          7272            DEC   C
    1F4B 20CC        7273            JR    NZ,TRANSP_10
    1F4D             7274 TRANSP_X
    1F4D C9          7275            RET
                     7276
                     7277
                     7278
    1F4E             7279 MIRROR_U_D                     ;REFLECT 8X8 PIXEL BLOCK AROUND THE
                     7280                                ; HORIZONTAL AXIS
                     7281
                     7282                                ; SOURCE IN HL, DESTINATION IN DE
                     7283                                ; DESTROYS AF,BC,DE,HL
                     7284
                     7285 * SOURCE  := SOURCE  + 7
    1F4E 010007      7286                LD              BC,7
    1F51 09          7287                ADD             HL,BC
                     7288
                     7289 * BYTE COUNT :=  8
    1F52 03          7290                INC             BC
                     7291
                     7292 * REPEAT
          <1F53>     7293 REFLECT_LOOP   EQU             $
                     7294
                     7295 *   [DESTINATION] := [SOURCE]
    1F53 7E          7296                LD              A,[HL]
    1F54 12          7297                LD              [DE],A
                     7298
                     7299 *    DESTINATION  := SUCC (DESTINATION)
    1F55 13          7300                INC             DE
                     7301
                     7302 *    SOURCE  := PRED (SOURCE)
    1F56 2B          7303                DEC             HL
                     7304
                     7305 *    BYTE COUNT := PRED BYTE COUNT
    1F57 0B          7306                DEC             BC
                     7307
                     7308 * UNTIL BYTE COUNT = 0
    1F58 78          7309                LD              A,B
    1F59 B1          7310                OR              C
    1F5A 20F7        7311                JR              NZ,REFLECT_LOOP
                     7312 * END
    1F5C C9          7313                RET
                     7314
                     7315                ;Modified February 14, 1983.  Filler locations were
                     7316                ;changed to OFFH to reflect OS_7PRIME.
                     7317
    1F5D FFFFFFFF    7318  HEX FF,FF,FF,FF             ;Filler
                     7319
                     7320
                     7321
```

```
XCATION OBJECT CODE LINE     SOURCE LINE

                    7324 ************************** ROM JUMP_TABLE ****************************
                    7325 ;
                    7326 ;JUMP_TABLE  THIS IS THE JUMP TABLE TO BE USED IN ACCESSING CODE
                    7327 ;            RESIDING IN THE O.S. ROM.  THIS TABLE MUST HAVE ITS
                    7328 ;            ORIGIN REDEFINED TO ACCOUNT FOR GROWTH. PILE NEW ROUTINES
                    7329 ;            AT THE BEGINNING OF THE TABLE MAKING SURE TO INCREMENT
                    7330 *            THE NO_OF_ROUTINES VALUE.
                    7331
                    7332 * NOTE ****
                    7333
                    7334 *               **** NO DELETIONS SHOULD BE MADE FROM ****
                    7335 *               **** THIS TABLE                       ****
                    7336
          <2000>    7337 ROM_END          EQU             2000H
                    7338 * THIS IS THE END OF OS ROM
                    7339
          <0035>    7340 NO_OF_ROUTINES  EQU 53
                    7341 * THIS NUMBER KEEPS COUNT OF THE NUMBER OF ROUTINES ACCESSED THROUGH
                    7342 * THE JUMP TABLE.
                    7343
                    7344 JUMP_TABLE      ORG  ROM_END-(NO_OF_ROUTINES*3)
                    7345
  1F61 C30300       7346 PLAY_SONGS      JP PLAY_SONGS_
  1F64 C30488       7347 ACTIVATEP       JP ACTIVATEQ
  1F67 C306C7       7348 PUTOBJP         JP PUTOBJQ
  1F6A C31D5A       7349 REFLECT_VERTICAL JP RFLCT_VERT
  1F6D C31D60       7350 REFLECT_HORIZONTAL JP RFLCT_HOR
  1F70 C31D66       7351 ROTATE_90       JP ROT_90
  1F73 C31D6C       7352 ENLARGE         JP ENLRG
  1F76 C3114A       7353 CONTROLLER_SCAN JP CONT_SCAN
  1F79 C31188       7354 DECODER         JP DECODER_
  1F7C C31979       7355 GAME_OPT        JP GAME_OPT_
  1F7F C31927       7356 LOAD_ASCII      JP LOAD_ASCII_
  1F82 C318D4       7357 FILL_VRAM       JP FILL_VRAM_
  1F85 C318E9       7358 MODE_1          JP MODE_1_
  1F88 C3116A       7359 UPDATE_SPINNER  JP UPDATE_SPINNER_
  1F8B C31B0E       7360 INIT_TABLEP     JP INIT_TABLEQ
  1F8E C31B8C       7361 GET_VRAMP       JP GET_VRAMQ
  1F91 C31C10       7362 PUT_VRAMP       JP PUT_VRAMQ
  1F94 C31C5A       7363 INIT_SPR_ORDERP JP INIT_SPR_ORDERQ
  1F97 C31C76       7364 WR_SPR_NM_TBLP  JP WR_SPR_NM_TBLQ
  1F9A C30F9A       7365 INIT_TIMERP     JP INIT_TIMERQ
  1F9D C30FB8       7366 FREE_SIGNALP    JP FREE_SIGNALQ
  1FA0 C31044       7367 REQUEST_SIGNALP JP REQUEST_SIGNALQ
  1FA3 C3108F       7368 TEST_SIGNALP    JP TEST_SIGNALQ
  1FA6 C31CBC       7369 WRITE_REGISTERP JP REG_WRITEQ
  1FA9 C31CED       7370 WRITE_VRAMP     JP VRAM_WRITEQ
  1FAC C31D2A       7371 READ_VRAMP      JP VRAM_READQ
  1FAF C30655       7372 INIT_WRITERP    JP INIT_QUEUEQ
  1FB2 C30203       7373 SOUND_INITP     JP INIT_SOUNDQ
  1FB5 C30251       7374 PLAY_ITP        JP JUKE_BOXQ
  1FB8 C3181D       7375 INIT_TABLE      JP INIT_TABLE_
  1FBB C318A3       7376 GET_VRAM        JP GET_VRAM_
  1FBE C31C27       7377 PUT_VRAM        JP PUT_VRAM_
  1FC1 C31C66       7378 INIT_SPR_ORDER  JP INIT_SPR_ORDER_
  1FC4 C31C82       7379 WR_SPR_NM_TBL   JP WR_SPR_NM_TBL_
  1FC7 C30FAA       7380 INIT_TIMER      JP INIT_TIMER_
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

    1FCA C30FC4      7381 FREE_SIGNAL     JP FREE_SIGNAL_
    1FCD C31053      7382 REQUEST_SIGNAL  JP REQUEST_SIGNAL_
    1FD0 C310CB      7383 TEST_SIGNAL     JP TEST_SIGNAL_
    1FD3 C30F37      7384 TIME_MGR        JP TIME_MGR_
    1FD6 C3023B      7385 TURN_OFF_SOUND  JP ALL_OFF
    1FD9 C31CCA      7386 WRITE_REGISTER  JP REG_WRITE,
    1FDC C31D57      7387 READ_REGISTER   JP REG_READ
    1FDF C31D01      7388 WRITE_VRAM      JP VRAM_WRITE
    1FE2 C31D3E      7389 READ_VRAM       JP VRAM_READ
    1FE5 C30664      7390 INIT_WRITER     JP INIT_QUEUE
    1FE8 C30679      7391 WRITER          JP WRITER_
    1FEB C311C1      7392 POLLER          JP POLLER_
    1FEE C30213      7393 SOUND_INIT      JP INIT_SOUND
    1FF1 C3025E      7394 PLAY_IT         JP JUKE_BOX
    1FF4 C3027F      7395 SOUND_MAN       JP SND_MANAGER
    1FF7 C304A3      7396 ACTIVATE        JP ACTIVATE_
    1FFA C306D8      7397 PUTOBJ          JP PUTOBJ_
    1FFD C30038      7398 RAND_GEN        JP RAND_GEN_
                     7399

Errors=    0
```

LINE#	SYMBOL	TYPE	REFERENCES
7396	ACTIVATE	A	242
7347	ACTIVATEP	A	248
1610	ACTIVATEQ	P	1609,7347
1627	ACTIVATE_	P	1587,1683,7396
1607	ACTIVATE_P	P	1611
1915	ACT_OSPRT	P	1648
1917	ACT_1SPRT	P	1650
1662	ACT_CMPLX	P	1652
1903	ACT_MOBILE	P	1646
1694	ACT_SEMI	P	1644
884	ADD816	P	703,883
5152	ADDR_ADJ	P	5172
5171	ADD_8	P	5152
6141	ADJUST_COUNT	P	6145
6130	ADJUST_INDEX	P	6134
3723	AD_EXIT	P	3719
3718	AD_LP	P	3721
463	AFTER_RANDOM	P	
85	ALEN	A	
1052	ALL_OFF	P	992,7385
6797	ALL_X	P	
492	AMERICA	P	236
87	APS	A	
88	APSV	A	733
964	AREA_SONG_IS	P	963,1216
4504	ARM	A	4856,5027
5008	ARM_DBNCE	P	4859
5037	ARM_EXIT	P	5022,5028,5033
4522	ARM_MASK	A	4746,5011
4516	ARM_OLD	A	5013,5021,5034
5024	ARM_REG	P	5020
5030	ARM_ST1	P	5016
4517	ARM_STATE	A	5014,5026,5036
499	ASCII_TABLE	P	254
5458	ASCII_TBL	P	499,5080
5382	ASC_TABLE	P	5681
86	ASTEP	A	1373,1445,1464
79	ATN	A	1370
724	ATN_SWEEP	P	721,1229
1038	B1	P	1040
6065	BASE_FACTORS	P	6049
3211	BK_CLR	A	3384,3413,3600,3658,3744,3833
3209	BK_PTN	A	3351,3413,3547,3650,3735,3744,3833
368	BOOT_UP	P	
1980	BUFFER	D	2021,2069,2129
7107	BYTE_COUNT	S	
2689	CALC_OFFSET	P	2480,2644
457	CARRY_READY	P	454
261	CARTRIDGE	A	260,534
6042	CASE_OF_CLR10	P	6039
6035	CASE_OF_COLOR	P	6023
6025	CASE_OF_GEN	P	6021
6032	CASE_OF_GEN10	P	6029
5705	CENTER_PRT	P	5234,5244
75	CH	A	
99	CHO	A	
90	CHOEND	A	

LINE#	SYMBOL	TYPE	REFERENCES
94	CHOREP	A	
100	CH1	A	
91	CH1END	A	
95	CH1REP	A	
101	CH2	A	
92	CH2END	A	
96	CH2REP	A	
102	CH3	A	
93	CH3END	A	
97	CH3REP	A	
4784	CHK_PLYR_1	P	4766,4780
4777	CHK_SEG_01	P	4772
4801	CHK_SEG_11	P	4796
4579	CINIT1	P	4587
1677	CMPLX4	P	1690
1691	CMPLX9	P	1676
2823	COLOR	A	
6337	COLORTABLE	D	5902,6330
2830	COLOR_AND_TAG	A	2936,3024,3064,3141
6670	COLOR_TABLE	A	6970,7059,7074
6997	COLOR_TEST	P	6817,6855,6903,6953
3201	COLR	A	3472,3602,3835
3559	COMBINE_LOOP	P	3593
3797	COM_PAT_COL	P	3576
3029	CONTINUE	P	2900,2988
6755	CONTINUE_GRAPHI	P	6690,6709,6730
4534	CONTROLLER_0	A	4611,4712
4535	CONTROLLER_1	A	
4570	CONTROLLER_INIT	P	555,4569
285	CONTROLLER_MAP	A	284,4573,4762
7353	CONTROLLER_SCAN	A	212
4609	CONT_READ	P	4720,4735
4615	CONT_READ1	P	4612
4617	CONT_READX	P	4614
4626	CONT_SCAN	P	4625,4760,7353
80	CTRL	A	1307
4531	CTRL_0_PORT	A	4613,4627,4635,4654
4530	CTRL_1_PORT	A	4616,4630,4638,4668
6432	CTRL_PORT	A	6433,6482,6487,6540,6544,6590,6594,6624
3594	C_LP_EXIT	P	3591
6431	DATA_PORT	A	6433,6550,6599
5050	DBNCE_BUFF	D	4576,4761
4092	DCR_L_MODE_TBL	P	
4104	DCR_L_RPT_TBL	P	4091
4129	DCR_S_MODE_TBL	P	4089
4086	DCR_TIMER	P	4072
833	DECLSN	P	689,735,741,832
850	DECMSN	P	849
7354	DECODER	A	213
4748	DECODERX	P	4728
4701	DECODER_	P	4700,7354
4820	DECODE_0	P	4776,4800
4842	DECODE_0X	P	4835
4854	DECODE_1	P	4783,4807
4866	DECODE_1X	P	4863
4827	DEC_FIRE	P	4823
4861	DEC_KBD	P	4857

LINE#	SYMBOL	TYPE	REFERENCES
4549	DEC_KBD_TBL	P	4739,4899
4715	DEC_PLYR	P	4713
4733	DEC_SEG1	P	4704
4833	DEC_SPNR	P	4829
59	DEDAREA	A	60,61,62,63,64,65
2181	DEFER	A	2184
630	DEFER_WRITES	D	238,559,2086,2091,2153,2183
4599	DELAY	P	4542
5723	DELAY_10	P	5258
7109	DESTINATION	S	
1535	DE_TO_DEST	P	1343,1441,1461,1487
5126	DISPLAY_LOGO	P	566,5082
5054	DIVIDE	P	6058
3304	DLP1	P	3324
3355	DLP2	P	3425
3479	DLP4	P	3542
3613	DLP5	P	3619
3638	DLP6	P	3642
4048	DONE	A	4473,4486
5167	DONE_LOGO	P	5147
1203	DONE_SNDMAN	P	1178
5120	DONT_PUT	P	2876,2879,2889,2892,2964,2967,2977,2980
2188	DO_PUTOBJ	P	2113,2185
1063	DUMAREA	P	992,1044,1168
1876	DUPLI	P	1878
3681	DVEX	P	3677
3676	DVLP	P	3680
1399	EFFECT	P	1342
1233	EFXOVER	P	1213
3705	ELSE04	P	3646
3223	ELSE1	P	3219
3530	ELSE10	P	3516
3608	ELSE13	P	3605
3810	ELSE18	P	3800
3282	ELSE2	P	3278
3841	ELSE23	P	3838
3339	ELSE5	P	3336
3400	ELSE6	P	3378
3463	ELSE8	P	3460
3501	ELSE9	P	3483
4227	ELSEZZ	P	6198,6201
2277	ELSE_1	P	2268
2702	ELSE_11	P	2694
2726	ELSE_12	P	2718
2536	ELSE_8	P	2503
2559	ELSE_9	P	2543,2544
3757	END04	P	3704
3225	END1	P	3221
3537	END10	P	3529
3587	END11	P	3584
3620	END12	P	3598
3610	END13	P	3607
3628	END14	P	3626
3750	END15	P	3728
3782	END16	P	3766
3781	END17	P	3773
3828	END18	P	3809

LINE#	SYMBOL	TYPE	REFERENCES
3815	END19	P	3813
3284	END2	P	3280
3821	END20	P	3819
3827	END21	P	3825
3872	END22	P	3831
3843	END23	P	3840
3852	END24	P	3847
3861	END25	P	3856
3870	END26	P	3865
3321	END3	P	3308
3313	END4	P	3311
3341	END5	P	3338
3420	END6	P	3399
3419	END7	P	3406
3466	END8	P	3462
3538	END9	P	3500
1389	ENDNOREP	P	1342,1382
1384	ENDREP	P	1342
73	ENDSDATA	A	1143,1177
6147	END_ADJ_COUNT	P	6140
6135	END_ADJ_INDEX	P	
376	END_BOOTUP	P	
6232	END_IFZZ	P	6225
2320	END_IF_1	P	2276
2610	END_IF_10	P	2584,2589
2707	END_IF_11	P	2698
2731	END_IF_12	P	2722
6828	END_IF_1_GRAPHI	P	6819
2344	END_IF_2	P	2326
6875	END_IF_2_GRAPHI	P	6857
2363	END_IF_3	P	2358
6914	END_IF_3_GRAPHI	P	6905
2419	END_IF_4	P	2395,2396
6982	END_IF_4_GRAPHI	P	6955
2572	END_IF_8	P	2532
2568	END_IF_9	P	2555
6611	END_INPUT	P	6607
6562	END_OUTPUT	P	6558
7352	ENLARGE	A	246
6746	ENLRG	P	6745,7352
6925	ENLRG_	P	6751
4051	EOT	A	4073,4215,4246,4407,4425,4464
2271	EQUAL_TO	P	2267
4305	EXIT	P	4264
7030	EXIT_FALSE	P	7018,7023
3155	EXIT_PUT_SPR	P	3118
7129	EXP_1	P	7134
7136	EXP_2	P	7141
526	FALSE	A	558,2090,7030
5616	FILL	P	5621
7357	FILL_VRAM	A	252,5780,5905
5610	FILL_VRAM_	P	5084,5129,7357
4501	FIRE	A	4828,4945
4926	FIRE_DBNCE	P	4831
4955	FIRE_EXIT	P	4940,4946,4951
4521	FIRE_MASK	A	4726,4929
4510	FIRE_OLD	A	4931,4939,4952

NE#	SYMBOL	TYPE	REFERENCES
,942	FIRE_REG	P	4938
,948	FIRE_ST1	P	4934
511	FIRE_STATE	A	4932,4944,4954
811	FIRST_GEN_NAME	A	3104
202	FLAGS	A	3225,3375,3596,3603,3644,3798,3829,3836
82	FPS	A	
83	FPSV	A	686
817	FRAME	A	2926,3014,3055,3092
814	FRAME_TABLE_PTR	A	2915,3003,3044,3081
050	FREE	A	4071,4222,4224,4248,4344,4432,4471,4483
214	FREE1	P	4219
237	FREE_COUNTER_	P	
311	FREE_EXIT	P	4216
221	FREE_MATCH	P	4212
309	FREE_SET	P	4223,4226,4228
381	FREE_SIGNAL	A	224
366	FREE_SIGNALP	A	229
200	FREE_SIGNALQ	P	4041,7366
206	FREE_SIGNAL_	P	4040,7381
197	FREE_SIG_PAR	P	4201
78	FREQ	A	807,811,814
671	FREQ_SWEEP	P	668,1230
203	FRM	A	3230,3441,3624
84	FSTEP	A	673,702,1372,1444
204	F_GEN	A	3269,3301,3623
333	GAME_NAME	A	332,5230,5232,5237,5247
355	GAME_OPT	A	250
777	GAME_OPT_	P	5761,7355
643	GET_BKGRND	P	2377,2642,3262
055	GET_COLOR	P	6822,6860,6908,6958
278	GET_NEXT	P	4249,4253,4258,4263,4277
294	GET_OLD	P	2291
376	GET_VRAM	A	189,3132,3380,3398,3411,3418
361	GET_VRAMP	A	194
077	GET_VRAMQ	P	5980,7361
087	GET_VRAM_	P	2659,5979,6776,7066,7376
074	GET_VRAM_P	P	6078
305	GRAPHICS	A	2913,2914,3001,3002,3042,3043,3079,3080,3102,3103
272	HEAD_ADDRESS	D	1997,2022,2032,2070
585	IF11	P	3582
71	INACTIVE	A	1063,1180,1217,1351,1389,1521
321	INIT_80	P	1822
058	INIT_QUEUE	P	2057,7390
050	INIT_QUEUEQ	P	2049,7372
045	INIT_QUEUE_P	P	2051
026	INIT_SOUND	P	992,7393
018	INIT_SOUNDQ	P	1011,7373
585	INIT_SOUND_DATA	D	1020,1022,1024
012	INIT_SOUND_PAR	P	1019
255	INIT_SPR10	P	6260
578	INIT_SPR_ORDER	A	191
563	INIT_SPR_ORDERP	A	196
242	INIT_SPR_ORDERQ	P	5980,7363
248	INIT_SPR_ORDER_	P	5979,7378
240	INIT_SPR_P	P	6243
575	INIT_TABLE	A	188,5649,5654,5659,5664,5669
046	INIT_TABLE80	P	6017,6024

LINE#	SYMBOL	TYPE	REFERENCES
6061	INIT_TABLE90	P	6031,6034,6041,6044
7360	INIT_TABLEP	A	193
5990	INIT_TABLEQ	P	5980,7360
5997	INIT_TABLE_	P	5979,7375
5987	INIT_TABLE_P	P	5991
7380	INIT_TIMER	A	223
7365	INIT_TIMERP	A	228
4174	INIT_TIMERQ	P	4039,7365
4180	INIT_TIMER_	P	4038,7380
4430	INIT_TIMER_EXI	P	4370,4396,4404
587	INIT_TIME_DATA	D	4176
4171	INIT_TIME_PAR	P	4175
7390	INIT_WRITER	A	203
7372	INIT_WRITERP	A	208
1807	INIT_XP_OS	P	1696,1905
6601	INPUT_LOOP	P	6605,6608
423	IRQ_INTERRUPT	P	
325	IRQ_INT_VECT	A	324,424
4502	JOY	A	4822,4986
4967	JOY_DBNCE	P	4825
4996	JOY_EXIT	P	4981,4987,4992
4523	JOY_MASK	A	4722,4970
4512	JOY_OLD	A	4972,4980,4993
4983	JOY_REG	P	4979
4989	JOY_ST1	P	4975
4513	JOY_STATE	A	4973,4985,4995
1102	JUKE_BOX	P	1075,1385,7394
1095	JUKE_BOXQ	P	1094,7374
1083	JUKE_BOX_PAR	P	1096
7344	JUMP_TABLE	A	
4505	KBD	A	4862,4903
4878	KBD_DBNCE	P	4865
4913	KBD_EXIT	P	4893,4904,4909
4520	KBD_MASK	A	4738,4882
4529	KBD_NULL	A	4550,4557,4561,4565
4518	KBD_OLD	A	4884,4892,4910
4895	KBD_REG	P	4891
4906	KBD_ST1	P	4887
4519	KBD_STATE	A	4885,4897,4912
1143	L1	P	1152
1229	L10	P	1222
1244	L12	P	1232,1240
1377	L13	P	1361
1394	L14	P	1379
1424	L15	P	1397
1448	L16	P	1429
1467	L17	P	1450
1482	L18	P	1479
1534	L19	P	
1176	L2	P	1202
684	L20	P	676
1528	L20_LOAD_NEX	P	1526
708	L21	P	691
766	L22	P	737,761
763	L23	P	743
793	L24	P	787
1306	L5	P	1302

LINE#	SYMBOL	TYPE	REFERENCES
4508	SEG_0	A	4771,4795
4509	SEG_1	A	4779,4803
1750	SEMI_BOT	P	1748
1755	SEMI_EXIT	P	1711,1753
1759	SEMI_GRI	P	1715
1745	SEMI_MID	P	1743
452	SET	P	446
6102	SET_COUNT	P	6097,6229
6124	SET_COUNT10	P	6118
6150	SET_COUNT20	P	6122,6129
6160	SET_COUNTX	P	
4145	SET_DONE_BIT	P	4103,4128,4136
2035	SET_UP_ENDIF	P	2024
1991	SET_UP_WRITE	P	2186
2824	SHAPE	A	
3575	SHFEX	P	3571
3570	SHFLP	P	3574
6164	SHIFT_CT	P	6125
4475	SIGNAL_FALSE	P	4465,4472
4470	SIGNAL_MATCH	P	4461
594	SIGNAL_NUM	D	4202,4204
4480	SIGNAL_TRUE	P	4474
4484	SIGNAL_TRUE1	P	4482
7199	SKIPZZ	P	7197
2317	SKIP_OLD	P	2293
1825	SM_BY_OLD	P	1814,1820
1138	SMD_MANAGER	P	1135,7395
76	SONGMO	A	
7393	SOUND_INIT	A	215
7373	SOUND_INITP	A	220
7395	SOUND_MAN	A	218
69	SOUND_PORT	A	795,810,821,1055,1057,1059,1061,1303,1324
7108	SOURCE	S	
5390	SPACE	P	5688
4503	SPIN	A	4834,4838,4839
4514	SPIN_OLD	A	
4515	SPIN_STATE	A	
5051	SPIN_SW0_CT	D	4589,4655,4710,4775
5052	SPIN_SW1_CT	D	4590,4799
4524	SPNR_MASK	A	
5334	SPRITEGENTBL	D	6327
5333	SPRITENAMETBL	D	6326
2807	SPRITE_INDEX	A	3112,3129,3147
275	SPRITE_ORDER	A	274,6254,6283
2798	SPRITE_PTR	S	
997	SR1ATN	A	1054,1278,1279
996	SR1FRQ	A	1280
999	SR2ATN	A	1056,1284,1285
998	SR2FRQ	A	1286
1001	SR3ATN	A	1058,1290,1291
1000	SR3FRQ	A	1292
1004	SRNATN	A	1060,1296,1297
1003	SRNCTL	A	1314
578	STACK	D	233,371
290	START_GAME	A	289,541,5266
2806	STATUS	A	2865,2866,2924,2925,2953,2954,3012,3013,3034,3035,3053,3054,3070,3071,3090,3091
4532	STRB_RST_PORT	A	4571,4641,4737

LINE#	SYMBOL	TYPE	REFERENCES
4533	STRB SET PORT	A	4633,4734
5053	STROBE_FLG	D	
4536	STROBE_RESET	A	
4537	STROBE_SET	A	4703
4266	SUBTRACT_4	P	4259
1831	SUP_GEN_CLR	P	1744,1749,1754
1894	SUP_UPDATE	P	1746,1751
2347	SV1	P	
2393	SV2	P	
573	SYSTEM_RAM_AREA	D	578
2260	S_OLD_SCRN	P	2255
1973	TAIL_ADDRESS	D	2071,2101,2130,2142
3981	TBL0	P	3978
588	TEMP1	D	4178
591	TEMP2	D	4179
4463	TEST1	P	4468
4489	TEST_EXIT	P	4478
7383	TEST_SIGNAL	A	226
7368	TEST_SIGNALP	A	231
4449	TEST_SIGNALQ	P	4045,7368
4455	TEST_SIGNAL_	P	4044,7383
601	TEST_SIG_NUM	D	4451,4453
4446	TEST_SIG_PARAM	P	4450
2802	THIS_SPRITE	S	
4343	TIMER1	P	4413,4428
4399	TIMER2	P	
5724	TIMER_1	P	5289,5732
5725	TIMER_2	P	5728
4147	TIMER_EXIT	P	4132,4157
598	TIMER_LENGTH	D	4334
4495	TIMER_TABLE_BAS	D	4030,4069,4181,4208,4243,4340,4457
7384	TIME_MGR	A	227
4067	TIME_MGRQ	P	4047
4068	TIME_MGR_	P	4046,7384
1319	TONE_OUT	P	1255,1282,1288,1294
5322	TRADEMARK	P	5190
7254	TRANSP_10	P	7273
7274	TRANSP_X	P	
527	TRUE	A	6818,6856,6904,6954,7026
7385	TURN_OFF_SOUND	A	216,548
1432	TYPE0	P	1342
1453	TYPE1	P	1342
1472	TYPE2	P	1342
1501	TYPE3	P	1342,1469
1625	T222	P	1623
783	UPATHCTRL	P	781,1306,1315,1327
4665	UPDATE_R0	P	4660
4679	UPDATE_R1	P	4674
4668	UPDATE_S1	P	4657,4663
7359	UPDATE_SPINNER	A	211
4653	UPDATE_SPINNER_	P	4652,7359
4681	UPDATE_SPINX	P	4670,4677
806	UPFREQ	P	805,1328
1166	UP_CH_DATA_PTRS	P	1121,1164,1241
608	VDP_MODE_WORD	D	235,1713,3216,6015,6120,6494,6502,7021
623	VDP_STATUS_BYTE	D	234
6352	VRAM_ADDR_TABLE	D	6009,6114,6286,6325

LINE#	SYMBOL	TYPE	REFERENCES
6586	VRAM_READ	P	2287,2316,6098,6585,7389
6577	VRAM_READQ	P	6576,7371
6572	VRAM_READ_P	P	6578
6529	VRAM_WRITE	P	1824,2417,6230,6528,7388
6520	VRAM_WRITEQ	P	6519,7370
6515	VRAM_WRITE_P	P	6521
1270	WAITE	A	
280	WORK_BUFFER	A	279,1873,2280,2361,2401,2859,2947,3215,3256,3268,3271,3291,3295,3300,3329,3347,3373,3382,3440,3451,3467 3471,3474,3545,3546,3599,3633,3649,3656,3657,3660,3703,3723,3737,3761,3774,3784,3785,6775,6805,6843,6863 6891,6929,6946,6961,6975,7047,7064,7079
7391	WRITER	A	204
2083	WRITER	P	2082,7391
5954	WRITE_CHAR	P	5844,5848,5852,5856,5948,5951
5959	WRITE_L11	P	5868,5871,5874,5877
5965	WRITE_L12	P	5889,5892,5895,5898
5941	WRITE_L3	P	5810,5813,5816,5819,5822,5825,5828,5831
5947	WRITE_L4	P	5834,5859,5880
5950	WRITE_L5	P	5837,5862,5883
5953	WRITE_L6	P	5840,5865,5886
5142	WRITE_LOOP	P	5165
5177	WRITE_NAMES	P	5169
7386	WRITE_REGISTER	A	199,5216,5263,5293,5638,5642,5674,5788,5912
7369	WRITE_REGISTERP	A	205
7388	WRITE_VRAM	A	201,3337
7370	WRITE_VRAMP	A	206
2135	WRTR_ELSE	P	2122
2145	WRTR_END_IF	P	2133
2149	WRTR_END_WHILE	P	2098
2094	WRTR_WHILE	P	2147
7379	WR_SPR_NM_TBL	A	192
7364	WR_SPR_NM_TBLP	A	197
6269	WR_SPR_NM_TBLQ	P	5980,7364
6275	WR_SPR_NM_TBL_	P	5979,7379
6267	WR_SPR_P	P	6270
2828	X	A	2910,2998,3039,3137
3200	XDISP	A	3239,3568
3208	XP_BK	A	3243,3260,3330,3681,3768,3788
2490	XP_NEG	P	2487
3206	XP_OS	A	3272,3762,3777
2498	X_IN_BOUNDS	P	
2818	X_LOCATION	A	2867,2955,3036
2827	Y	A	3075
3199	YDISP	A	3248,3549,3577,3579
3207	YP_BK	A	3252,3257,3259,3292,3348,3374,3634,3661,3685,3724,3769,3786,3789
3205	YP_OS	A	3296,3767,3775,3778
2819	Y_LOCATION	A	3072