```
LOCATION OBJECT CODE LINE     SOURCE LINE

  073B C5           2262          PUSH BC          ;SAVE REGS
  073C D5           2263          PUSH DE
  073D E5           2264          PUSH HL
                    2265
  073E FE70         2266                 CP               70H
  0740 2802         2267                 JR               Z,EQUAL_TO
  0742 3807         2268                 JR               C,ELSE_1
                    2269
                    2270 ;        IF [.A,GE,70H]        ;THEN OLD_SCREEN IN CPU RAM
  0744              2271 EQUAL_TO
  0744 67           2272           LD H,A
  0745 DD6E04       2273           LD L,[IX+4]           ;HL := OLD_SCREEN ADDRESS
  0748 7E           2274           LD A,[HL]
                    2275
  0749 1835         2276                 JR               END_IF_1
  074B              2277 ELSE_1
                    2278
                    2279 ;        ELSE                  ;OLD_SCREEN IN VRAM
  074B 2A8006       2280           LD HL,[WORK_BUFFER]   ;GET ADDRESS OF FREE BUFFER SPACE
  074E DD5605       2281           LD D,[IX+5]           ;DE := OLD_SCREEN ADDRESS
  0751 DD5E04       2282           LD E,[IX+4]
  0754 E5           2283           PUSH HL               ;SAVE 2 COPIES FREE BUFFER ADDR
  0755 D5           2284           PUSH DE               ;SAVE OLD_SCREEN ADDR
  0756 E5           2285           PUSH HL
  0757 010004       2286           LD BC,4               ;READ 4 BYTES [X,Y_PAT_POSs, X,Y_EXTETNs]
  075A CD1D3E       2287           CALL VRAM_READ
  075D E1           2288           POP HL                ;HL := FREE BUFFER ADDR
  075E 7E           2289           LD  A,[HL]
  075F FE80         2290           CP  80H
  0761 2003         2291           JR  NZ,GET_OLD
  0763 D1           2292           POP  DE
  0764 1819         2293           JR  SKIP_OLD
  0766 23           2294 GET_OLD   INC HL
  0767 23           2295           INC HL
  0768 46           2296           LD B,[HL]             ;B := X_EXTENT OF OLD_SCREEN
  0769 23           2297           INC HL
  076A 5E           2298           LD E,[HL]             ;E := Y_EXTENT
  076B 1600         2299           LD D,0
  076D 23           2300           INC HL
  076E EB           2301           EX DE,HL              ;MULTIPLY X_EXTENT*Y_EXTENT IN HL
                    2302
  076F 1801         2303                 JR               M_XY+1
  0771 29           2304 M_XY:           ADD              HL,HL
  0772 10FD         2305                 DJNZ             M_XY
                    2306
                    2307 ;         DJNZ M.XY
  0774 E5           2308           PUSH HL
  0775 C1           2309           POP BC                ;BC := NUMBER OF BYTES TO READ
  0776 EB           2310           EX DE,HL              ;HL := FREE BUFF ADDR + 4
  0777 D1           2311           POP DE                ;DE := OLD_SCREEN ADDR.
  0778 13           2312           INC DE
  0779 13           2313           INC DE
  077A 13           2314           INC DE
  077B 13           2315           INC DE
  077C CD1D3E       2316           CALL VRAM_READ         ;READ SAVED NAMES FOR BACKGROUND
  077F E1           2317 SKIP_OLD  POP HL                ;HL := FREE BUFF ADDR.
                    2318 ;        ENDIF
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                    2319
  0780              2320 END_IF_1
                    2321
  0780 7E           2322          LD A,[HL]                ;A := X_PAT_POS
                    2323
                    2324
  0781 FE80         2325                  CP               80H
  0783 280F         2326                  JR               Z,END_IF_2
                    2327
                    2328 ;        IF [.A,NE,80H]           ;THEN THERE IS AN OLD_SCREEN
  0785 5E           2329            LD E,[HL]                ;E := X_PAT_POS
  0786 23           2330            INC HL
  0787 56           2331            LD D,[HL]                ;D := Y_PAT_POS
  0788 23           2332            INC HL
  0789 4E           2333            LD C,[HL]                ;C := X_EXTENT
  078A 23           2334            INC HL
  078B 46           2335            LD B,[HL]                ;B := Y_EXTENT
  078C 23           2336            INC HL                   ;HL POINTS TO FIRST NAME IN LIST
                    2337
  078D DDE5         2338                  PUSH             IX              ;SAVE OBJECT POINTER
                    2339
  078F CD0808       2340            CALL PUTFRAME            ;RESTORE OLD_SCREEN TO DISPLAY
                    2341
  0792 DDE1         2342                  POP              IX              ;RESTORE OBJECT POINTER
                    2343
  0794              2344 END_IF_2
                    2345
                    2346 ;        ENDIF
  0794 E1           2347 SV1:     POP HL                   ;HL := ADDR OF FIRST NAME IN FRAME
  0795 D1           2348          POP DE                   ;DE := Y,X_PAT_POSs
  0796 C1           2349          POP BC                   ;BC := Y,X_EXTENTs
  0797 C5           2350          PUSH BC
  0798 D5           2351          PUSH DE
  0799 E5           2352          PUSH HL
  079A DD6605       2353          LD H,[IX+5]              ;HL := OLD_SCREEN ADDRESS
  079D DD6E04       2354          LD L,[IX+4]
                    2355
  07A0 3E70         2356                  LD               A,70H
  07A2 BC           2357                  CP               H
  07A3 3803         2358                  JR               C,END_IF_3
                    2359
                    2360 ;        IF [.H,LT,70H]           ;THE OLD_SCREEN NOW IN FREE BUFFER
  07A5 2A8006       2361            LD HL,[WORK_BUFFER]      ;THEREFORE, MOVE BACKGROUND TO BUFFER
                    2362
  07A8              2363 END_IF_3
                    2364
                    2365 ;        ENDIF
  07A8 73           2366          LD [HL],E                ;OLD_SCREEN +   0 := X_PAT_POS
  07A9 23           2367          INC HL
  07AA 72           2368          LD [HL],D                ;      "        1 := Y_PAT_POS
  07AB 23           2369          INC HL
  07AC 71           2370          LD [HL],C                ;      "        2 := X_EXTENT
  07AD 23           2371          INC HL
  07AE 70           2372          LD [HL],B                ;      "        3 := Y_EXTENT
  07AF 23           2373          INC HL                   ;HL := ADDRESS TO STORE NAMES
                    2374
  07[illegible]      5  2375                  POS                                    OLD.H     INTI
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     2376
   07B2 CD0898       2377          CALL GET_BKGRND
                     2378
   07B5 DDE1         2379                  POP             IX                 ;RESTORE OBJECT POINTER
                     2380
   07B7 E1           2381          POP HL                  ;WHERE NAMES ARE IN CPU RAM
   07B8 D1           2382          POP DE                  ;WHERE TO MOVE THEM TO IN VRAM [NAME TABLE]
   07B9 C1           2383          POP BC                  ;HOW MANY TO MOVE
                     2384
   07BA DDE5         2385                  PUSH            IX                 ;SAVE OBJECT POINTER
                     2386
   07BC CD0808       2387          CALL PUTFRAME
                     2388
   07BF DDE1         2389                  POP             IX                 ;RESTORE OBJECT POINTER
                     2390
   07C1 DD5605       2391          LD D,[IX+5]             ;SEE IF SAVED BACKGROUND TO BE MOVED TO VRAM
                     2392
   07C4 3E70         2393 SV2:             LD              A,70H
   07C6 BA           2394                  CP              D
   07C7 281E         2395                  JR              Z,END_IF_4
   07C9 381C         2396                  JR              C,END_IF_4
                     2397
                     2398 ;SV2::    IF [.D,LT,70H]
   07CB DD5E04       2399             LD E,[IX+4]          ;DE := OLD_SCREEN ADDR
   07CE D9           2400             EXX                  ;USE 'REG FOR CALCULATION
   07CF 2A8006       2401             LD HL,[WORK_BUFFER]  ;WHERE NEXT OLD_SCREEN DATA IS
   07D2 E5           2402             PUSH HL
   07D3 23           2403             INC HL
   07D4 23           2404             INC HL
   07D5 5E           2405             LD E,[HL]            ;E := X_EXTENT
   07D6 1600         2406             LD D,0
   07D8 23           2407             INC HL
   07D9 46           2408             LD B,[HL]            ;B := Y_EXTENT
   07DA EB           2409             EX DE,HL             ;HL := X_EXTENT
   07DB 1801         2410             JR M_XY2+1
   07DD 29           2411 M_XY2:      ADD HL,HL            ;HL := X_EXTENT*Y_EXTENT
   07DE 10FD         2412             DJNZ M_XY2
   07E0 E5           2413             PUSH HL
   07E1 D9           2414             EXX
   07E2 C1           2415             POP BC               ;BC := NUMBER OF BYTES TO WRITE
   07E3 E1           2416             POP HL               ;HL := FREE BUFFER ADDRESS
   07E4 CD1D01       2417             CALL VRAM_WRITE
                     2418
   07E7              2419 END_IF_4
                     2420
                     2421 ;         ENDIF
   07E7 C9           2422          RET
                     2423
                     2424 ******************* PX.TO.PTRN.POS ******************************************
                     2425 ;DESCRIPTION:     DIVIDES REG DE BY 8, IF SIGNED RESULT > 127 THEN E := MAX SIGNED
                     2426 ;                 POSITIVE NUMBER. IF RESULT < -128, THEN E := MIN NEGATIVE NUM
                     2427 ;INPUT:             DE = 16 BIT SIGNED NUMBER
                     2428 ;OUTPUT:            DE/8 < -128      E = -128
                     2429 ;           -128 <= DE/8 <=+127      E = DE/8
                     2430 ;           +127 <  DE/8             E = +127
                     2431 ****************************************************************************
                     2432 ;
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     2433
                     2434                GLB             PX_TO_PTRN_POS
  07E8               2435 PX_TO_PTRN_POS
                     2436
  07E8 E5            2437         PUSH HL                ;HL USED TO TEST MAGNITUDE
  07E9 CB2A          2438         SRA D                  ;16 BIT SHIFT LEFT
  07EB CB1B          2439         RR E
  07ED CB2A          2440         SRA D
  07EF CB1B          2441         RR E
  07F1 CB2A          2442         SRA D                  ;                X3
  07F3 CB1B          2443         RR E
  07F5 CB7A          2444         BIT 7,D                ;IS RESULT NEGATIVE
  07F7 2009          2445         JR NZ,NEGTV
  07F9 21FF80        2446         LD HL,0FF80H           ;SEE IF RESULT < 127
  07FC 19            2447         ADD HL,DE
  07FD E1            2448         POP HL
  07FE D0            2449         RET NC
  07FF 1E7F          2450         LD E,7FH               ;IF > 128, THEN E := MAX SIGNED + NUM.
  0801 C9            2451         RET
                     2452 ;XXXXX
  0802 210080        2453 NEGTV:  LD HL,080H             ;IS RESULT > -128
                     2454 ;XXXXX
  0805 19            2455         ADD HL,DE
  0806 E1            2456         POP HL
  0807 D8            2457         RET C
  0808 1E80          2458         LD E,80H               ;IF < -128, THE E := MIN SIGNED - NUM.
  080A C9            2459         RET
                     2460 ;
                     2461 ************************** PUT FRAME *******************************************
                     2462 ;DESCRIPTION:    THE NAMES WHICH CONSTITUTE A FRAME ARE MOVED TO THE NAME TABLE
                     2463 ;                IN VRAM.  THE UPPER LEFT HAND CORNER OF THE FRAME IS POSITIONED
                     2464 ;                AT X_PAT_POS, Y_PAT_POS.
                     2465 ;INPUT:           HL = ADDRESS OF LIST OF NAMES (IN CPU RAM)
                     2466 ;                 E = X_PAT_POS
                     2467 ;                 D = Y_PAT_POS
                     2468 ;                 C = X_EXTENT
                     2469 ;                 B = Y_EXTENT
                     2470 ******************************************************************************
                     2471 ;
                     2472                GLB             PUTFRAME
  080B C5            2473 PUTFRAME:  PUSH BC             ;COPY PARAMETERS INTO PRIMED REGISTERS
  080C D5            2474         PUSH DE
  080D E5            2475         PUSH HL                ;AND FRAME ADDRESS INTO DE'
  080E D9            2476         EXX
  080F E1            2477         POP HL
  0810 D1            2478         POP DE
  0811 C1            2479         POP BC
  0812 CD08C0        2480         CALL CALC_OFFSET
  0815 D9            2481         EXX
                     2482 ;XXXXX
                     2483 ; TEST FOR THE FOLLOWING CONDITION: (X_PAT_POS SLE 32) AND (X_PAT_POS + X_EXTENT
                     2484 ;                                                     SGT 0)
  0816 7B            2485 PF1:    LD A,E                 ;IS X_PAT_POS < 0?
  0817 CB7F          2486         BIT 7,A
  0819 2003          2487         JR NZ,XP_NEG
  081B FE20          2488         CP 32                  ;IS X_PAT_POS < 32?
  081D [illegible]   [illegible]  RET [illegible]        [illegible]
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  081E 81          2490 XP_NEG: ADD A,C               ;A := X_PAT_POS + X_EXTENT
  081F CB7F        2491         BIT 7,A               ;IS A NEG?
  0821 C0          2492         RET NZ                ;YES, RETURN
  0822 B7          2493         OR A                  ;A = 0?
  0823 C8          2494         RET Z                 ;RETURN IF 0
                   2495 ;XXXXX
                   2496 ;X.IN.BOUNDS::
                   2497
  0824             2498 X_IN_BOUNDS
                   2499
                   2500 ;        IF [.E,IS,MINUS]      ;IF X_PAT_POS < 0 , FRAME BLEEDING ON FROM LEFT
                   2501
  0824 CB7B        2502                 BIT           7,E
  0826 2820        2503                 JR            Z,ELSE_8
                   2504
  0828 79          2505           LD A,C                ;CALCULATE AMOUNT OF FRAME ON SCREEN
  0829 83          2506           ADD A,E               ;A := X_EXTENT + X_PAT_POS
  082A D5          2507           PUSH DE
                   2508 ;XXXXX
  082B FE21        2509           CP 33                 ;IF NUMBER OF NAMES > 32
  082D 3802        2510           JR C,LT33
  082F 3E20        2511           LD A,32               ;THEN NUMBER OF NAMES := 32
  0831 5F          2512 LT33:     LD E,A                ;NUMBER OF NAMES ON SCREEN EACH ROW
                   2513 ;XXXXX
  0832 1600        2514           LD D,0
  0834 D5          2515           PUSH DE               ;GET COUNT INTO IY
  0835 FDE1        2516           POP IY
  0837 D1          2517           POP DE                ;RESTORE DE
  0838 7B          2518           LD A,E                ;A := X_PAT_POS
  0839 D9          2519           EXX                   ;NOW ADJUST STARTING POINTS IN FRAME LIST AND
                   2520                                 ;NAME TABLE
  083A C5          2521           PUSH BC               ;SAVE X AND Y EXTENT
  083B ED44        2522           NEG                   ;2'S COMPLIMENT OF X_PAT_POS
  083D 4F          2523           LD C,A
  083E 0600        2524           LD B,0
  0840 09          2525           ADD HL,BC             ;ADD DISPLACEMENT TO FRAME POINTER
  0841 EB          2526           EX DE,HL
  0842 09          2527           ADD HL,BC             ;ADD DISPLACEMENT TO NAME TABLE POINTER
  0843 EB          2528           EX DE,HL
  0844 C1          2529           POP BC
  0845 D9          2530           EXX
                   2531
  0846 181C        2532                 JR            END_IF_8
                   2533
                   2534 ;       ELSE
                   2535
  0848             2536 ELSE_8
                   2537
  0848 7B          2538 PF2:     LD A,E                ;IS X_PAT_POS + X_EXTENT > 31
  0849 81          2539          ADD A,C
                   2540 ;        IF [.A,GT,31]
                   2541
  084A FE1F        2542                 CP            31
  084C 280F        2543                 JR            Z,ELSE_9
  084E 3800        2544                 JR            C,ELSE_9
                   2545
  0850 3E20        2546            LD A,32              ;SUBTRACT X_PAT_POS FROM 31
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  0852 93            2547                SUB E
  0853 D5            2548                PUSH DE
  0854 5F            2549                LD E,A                  ;GET THIS NUMBER INTO IY
  0855 1600          2550                LD D,0
  0857 D5            2551                PUSH DE
  0858 FDE1          2552                POP IY
  085A D1            2553                POP DE
                     2554
  085B 1807          2555                    JR              END_IF_9
                     2556
                     2557 ;           ELSE                    ;BOTH ENDS OF FRAME WITHIN PATTERN PLANE
                     2558
  085D               2559 ELSE_9
                     2560
  085D C5            2561 PF3:           PUSH BC
  085E 0600          2562                LD B,0
  0860 C5            2563                PUSH BC
  0861 FDE1          2564                POP IY
  0863 C1            2565                POP BC
                     2566 ;           ENDIF
                     2567
  0864               2568 END_IF_9
                     2569
                     2570 ;         ENDIF
                     2571
  0864               2572 END_IF_8
                     2573
  0864 1E00          2574            LD E,0
                     2575 ;         REPEAT                   ;Y_EXTENT-1 TIMES
                     2576
  0866               2577 RPT_1
                     2578
  0866 7A            2579 PF4:       LD A,D                   ;GET Y_PAT_POS
  0867 83            2580            ADD A,E                  ;ADD Y
                     2581 ;         IF [.A,IS,PLUS]
                     2582
  0868 CB7F          2583                BIT             7,A
  086A 2019          2584                JR              NZ,END_IF_10
                     2585
                     2586 ;             IF [.A,LE,23]           ;IS 0<=Y_PAT_POS + Y <=23
                     2587
  086C FE18          2588                    CP          24
  086E 3015          2589                    JR          NC,END_IF_10
                     2590
  0870 C5            2591                  PUSH BC
  0871 D5            2592                  PUSH DE
  0872 D9            2593                  EXX
  0873 C5            2594                  PUSH BC
  0874 D5            2595                  PUSH DE
  0875 E5            2596                  PUSH HL
  0876 FDE5          2597                  PUSH IY
  0878 3E02          2598                  LD A,2                  ;CODE FOR PATTERN NAME TABLE ADDED 4/20
  087A CD1C27        2599                  CALL PUT_VRAM_
  087D FDE1          2600                  POP IY
  087F E1            2601                  POP HL
  0880 D1            2602                  POP DE
  0881 C1            2603                  POP BC
```

```
OCATION OBJECT CODE LINE     SOURCE LINE

  0882 D9          2604                 EXX
  0883 D1          2605                 POP DE
  0884 C1          2606                 POP BC
                   2607 ;             ENDIF
                   2608 ;           ENDIF
                   2609
  0885             2610 END_IF_10
                   2611
  0885 D9          2612             EXX
  0886 C5          2613             PUSH BC
  0887 0600        2614             LD B,0                  ;INCREMENT POINTER INTO FRAME BY X_EXTENT
  0889 09          2615             ADD HL,BC
  088A EB          2616             EX DE,HL
  088B 010020      2617             LD BC,32                ;INCREMENT OFFSET BY 32
  088E 09          2618             ADD HL,BC
  088F EB          2619             EX DE,HL
  0890 C1          2620             POP BC
  0891 D9          2621             EXX
  0892 1C          2622             INC E
                   2623 ;         UNTIL [.E,EQ,.B]        ;UNTIL Y=Y_EXTENT
                   2624
  0893 7B          2625                     LD              A,E
  0894 B8          2626                     CP              B
  0895 20CF        2627                     JR              NZ,RPT_1
                   2628
  0897 C9          2629             RET
                   2630 ;
                   2631 ;           .COMMENT }
                   2632 *************** GET.BKGRND ******************************************************
                   2633 ;DESCRIPTION:     THIS ROUTINE GETS THE NAMES FROM THE NAME TABLE WHICH CONSTITUTE
                   2634 ;                 THE BACKGROUND ON WHICH AN OBJECT IS TO BE MOVED
                   2635 ;INPUT:           HL = LOCATION IN CPU RAM TO WHICH THE NAMES ARE MOVED
                   2636 ;                 D = Y_PAT_POS [TOP ROW OF PATTERN]
                   2637 ;                 E = X_PAT_POS [LEFT HAND COLUMN]
                   2638 ;                 B = Y_EXTENT OF PATTERN
                   2639 ;                 C = X_EXTENT OF PATTERN
                   2640 ********************************************************************************
                   2641 ;
                   2642                     GLB             GET_BKGRND
  0898             2643 GET_BKGRND:
  0898 CD08C0      2644             CALL CALC_OFFSET        ;OFFSET INTO NAME TABLE OF POSITION OF UPPER LEFT
  089B C5          2645             PUSH BC                 ;                                   HAND PATTERN
  089C 0600        2646             LD B,0                  ;GET X_EXTENT INTO IY
  089E C5          2647             PUSH BC                 ;   NUMBER OF NAMES PER ROW
  089F FDE1        2648             POP IY
  08A1 C1          2649             POP BC
                   2650 ;           REPEAT                  ;Y_EXTENT-1 TIMES
                   2651
  08A2             2652 RPT_2
                   2653
  08A2 C5          2654              PUSH BC
  08A3 D5          2655              PUSH DE
  08A4 E5          2656              PUSH HL
  08A5 FDE5        2657              PUSH IY
  08A7 3E02        2658              LD   A,2               ;TABLE CODE FOR PATTERN NAME TABLE
  08A9 CD1BA3      2659              CALL GET_VRAM_
  08AC FDE1        2660              POP IY
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                    2747
                    2748 * THIS MODULE CONTAINS CODE FOR THE PUT1SPRITE AND PUT0SPRITE
                    2749 * ROUTINES. THESE ROUTINES TURN OUT TO BE ESSENTIALLY THE SAME CODE
                    2750 * WITH TWO SLIGHTLY DIFFERENT ENTRY POINTS
                    2751
                    2752
                    2753 * IT IS CALLED WITH THE ADDRESS OF THE SPRITE OBJECT IN THE IX REGISTER.
                    2754
                    2755 * THE FORMAT FOR SPRITE OBJECTS IS
                    2756
                    2757 * SPRITE_OBJECT = RECORD
                    2758 *   GRAPHICS:^SPRITE_GRAPHICS
                    2759 *   STATUS:^SPRITE_STATUS
                    2760 *   SPRITE_INDEX:BYTE             (SPRITE_NAME_TABLE INDEX OF THIS SPRITE)
                    2761 * END SPRITE_OBJECT
                    2762
                    2763 * SPRITE_GRAPHICS = RECORD
                    2764 *   OBJECT_TYPE:BYTE
                    2765 *   FIRST_GEN_NAME:BYTE           (NAME OF FIRST SPRITE GENERATOR)
                    2766 *   PTRN_POINTER:^PATTERN_GENERATOR   (POINTER TO ROM'ED GENERATORS)
                    2767 *   NUMGEN:BYTE                   (NUMBER OF ROM'ED GENERATORS)
                    2768 *   FRAME_TABLE_PTR:^ARRAY[0..nn] OF FRAME (TABLE OF ANIMATION FRAMES)
                    2769 * END SPRITE_ROM_GRAPHICS
                    2770
                    2771 * SPRITE_STATUS  = RECORD
                    2772 *   FRAME:BYTE            (CURRENT ANIMATION FRAME)
                    2773 *   X_LOCATION:INTEGER
                    2774 *   Y_LOCATION:INTEGER
                    2775 *   NEXT_GEN:BYTE         (INDEX OF FREE SPACE IN GENERATOR TABLE)
                    2776 * END SPRITE_STATUS
                    2777
                    2778 * FRAME = RECORD
                    2779 *   COLOR:BYTE            (SPRITE'S COLOR FOR THIS FRAME)
                    2780 *   SHAPE:BYTE            (THIS FRAME'S OFFSET FROM NAME FROM FIRST_GEN_NAME)
                    2781 * END FRAME
                    2782
                    2783 * SPRITE = RECORD
                    2784 *   Y:BYTE
                    2785 *   X:BYTE
                    2786 *   NAME:BYTE
                    2787 *   COLOR_AND_TAG:BYTE
                    2788 * END SPRITE
                    2789
                    2790 ***************************************************************************
                    2791
                    2792 *************************** DICTIONARY *****************************
                    2793
                    2794 ;                EXT             WORK_BUFFER
                    2795 * WORK_BUFFER IS A POINTER IN CARTRIDGE ROM, LOCATED AT 8006H, TO THE
                    2796 * FREE BUFFER AREA TO BE USED BY THE GRAPHICS ROUTINES.
                    2797
        <000F>      2798 SPRITE_PTR       EQU             IY
                    2799 * SPRITE_PTR IS A POINTER TO THE NEW SPRITE NAME TABLE ENTRY BEING
                    2800 * BUILT BY THIS ROUTINE.
                    2801
        <0          2802      SPR         EQU             IX
                    2803 * THIS SPRITE IS A POINTER TO THE SPRITE OBJECT BEING PUT
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     2804
           <0000>    2805 GRAPHICS        EQU             0
           <0002>    2806 STATUS          EQU             2
           <0004>    2807 SPRITE_INDEX    EQU             4
                     2808 * FIELD OFFSETS FOR SPRITE_OBJECT RECORDS
                     2809
           <0000>    2810 OBJECT_TYPE     EQU             0
           <0001>    2811 FIRST_GEN_NAME  EQU             1
           <0002>    2812 PTRN_POINTER    EQU             2
           <0004>    2813 NUMGEN          EQU             4
           <0005>    2814 FRAME_TABLE_PTR EQU             5
                     2815 * FIELD OFFSETS FOR SPRITE GRAPHICS RECORDS
                     2816
           <0000>    2817 FRAME           EQU             0
           <0001>    2818 X_LOCATION      EQU             1
           <0003>    2819 Y_LOCATION      EQU             3
           <0005>    2820 NEXT_GEN        EQU             5
                     2821 * FIELD OFFSETS FOR SPRITE_STATUS RECORDS
                     2822
           <0000>    2823 COLOR           EQU             0
           <0001>    2824 SHAPE           EQU             1
                     2825 * FIELD OFFSETS FOR FRAME RECORDS
                     2826
           <0000>    2827 Y               EQU             0
           <0001>    2828 X               EQU             1
           <0002>    2829 NAME            EQU             2
           <0003>    2830 COLOR_AND_TAG   EQU             3
                     2831 * FIELD OFFSETS FOR SPRITE RECORDS
                     2832 *********************** EXTERNAL PROCEDURES ************************
                     2833
                     2834 ; EXT PUT_VRAM,GET_VRAM
                     2835 * EXTERNAL PROCEDURE PUT_VRAMP (TABLE_CODE:BYTE; START_INDEX,SLICE:BYTE;
                     2836 *                               VAR DATA:BUFFER;ITEM_COUNT:INTEGER);
                     2837
                     2838 * EXTERNAL PROCEDURE GET_VRAMP (TABLE_CODE:BYTE; START_INDEX,SLICE:BYTE;
                     2839 *                               VAR DATA:BUFFER;ITEM_COUNT:INTEGER);
                     2840
                     2841 * PUT_VRAM SENDS A BLOCK OF DATA TO THE TABLE SPECIFIED BY TABLE_CODE.
                     2842 * THE SLICE, START_INDEX, AND ITEM_COUNT ARE TABLE DEPENDANT. GET_VRAM
                     2843 * DOES THE INVERSE OPERATION.
                     2844
                     2845 *    - TABLE_CODE IS PASSED IN A
                     2846 *    - START_INDEX,SLICE IN DE
                     2847 *    - DATA BUFFER ADDRESS IN HL
                     2848 *    - BYTE COUNT PASSED IN IY
                     2849
                     2850 ************************ PROCEDURE BODY ***************************
                     2851
                     2852                 PROG
                     2853  GLB PUT0SPRITE,PUT1SPRITE
                     2854
                     2855 * BEGIN PUT0SPRITE
           <080F>    2856 PUT0SPRITE      EQU             $
                     2857
                     2858 *    SPRITE_PTR := WORK_BUFFER
 080F FD2A8006       2859                 LD              SPRITE_PTR,[WORK_BUFFER]
                     2860
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     2861 *   WITH THIS_SPRITE^,SPRITE_PTR^ DO
                     2862
                     2863 *     IF (STATUS^.X_LOCATION > -8) AND (STATUS^.X_LOCATION < 256) AND
                     2864 *        (STATUS^.Y_LOCATION > -8) AND (STATUS^.Y_LOCATION < 192) THEN
  08E3 DD6E02        2865               LD          L,[THIS_SPRITE+STATUS]
  08E6 DD6603        2866               LD          H,[THIS_SPRITE+STATUS+1]
  08E9 110001        2867               LD          DE,X_LOCATION
  08EC 19            2868               ADD         HL,DE           ; [HL] = X_LOCATION
  08ED 4E            2869               LD          C,[HL]
  08EE 23            2870               INC         HL
  08EF 46            2871               LD          B,[HL]          ; BC = X_LOCATION
  08F0 78            2872               LD          A,B             ; COMPARE BC WITH -8
  08F1 FE00          2873               CP          0
  08F3 2808          2874               JR          Z,OK__1
  08F5 FEFF          2875               CP          -1
  08F7 C20A54        2876               JP          NZ,DONT_PUT
  08FA 79            2877               LD          A,C
  08FB FEF9          2878               CP          -7
  08FD FA0A54        2879               JP          M,DONT_PUT
  0900               2880 OK__1
  0900 23            2881               INC         HL              ; [HL] = Y_LOCATION
  0901 4E            2882               LD          C,[HL]
  0902 23            2883               INC         HL
  0903 46            2884               LD          B,[HL]          ; BC = Y_LOCATION
  0904 78            2885               LD          A,B             ; COMPARE BC WITH -8
  0905 FE00          2886               CP          0
  0907 2808          2887               JR          Z,OK__2
  0909 FEFF          2888               CP          -1
  090B C20A54        2889               JP          NZ,DONT_PUT
  090E 79            2890               LD          A,C
  090F FEF9          2891               CP          -7
  0911 FA0A54        2892               JP          M,DONT_PUT
  0914               2893 OK__2
                     2894
                     2895 *      IF STATUS^.X_LOCATION < 0 THEN
  0914 2B            2896               DEC         HL
  0915 2B            2897               DEC         HL              ; [HL] = HI(X_LOCATION)
  0916 7E            2898               LD          A,[HL]          ; COMPARE WITH 0
  0917 FE00          2899               CP          0
  0919 CA09CA        2900               JP          Z,CONTINUE
                     2901
                     2902 *        X := BYTE(STATUS^.X_LOCATION) + 8
  091C 2B            2903               DEC         HL              ;[HL] = ^X_LOCATION
  091D 4E            2904               LD          C,[HL]
  091E 23            2905               INC         HL
  091F 46            2906               LD          B,[HL]
  0920 210008        2907               LD          HL,8
  0923 09            2908               ADD         HL,BC
  0924 7D            2909               LD          A,L
  0925 FD7701        2910               LD          [SPRITE_PTR+X],A
                     2911
                     2912 *        COLOR_AND_TAG := GRAPHICS^.FRAME_TABLE[STATUS^.FRAME].COLOR OR 80H
  0928 DD6E00        2913               LD          L,[THIS_SPRITE+GRAPHICS]
  092B DD6601        2914               LD          H,[THIS_SPRITE+GRAPHICS+1]
  092E 110005        2915               LD          DE,FRAME_TABLE_PTR
  0931 19            2916               ADD         HL,DE           ;[HL] = FRAME_TABLE_PTR
  0932 [illegible]
```

```
LOCATION OBJECT CODE LINE      SOURCE LINE

   0933 1A             2918                  LD          A,[DE]
   0934 6F             2919                  LD          L,A
   0935 13             2920                  INC         DE
   0936 1A             2921                  LD          A,[DE]
   0937 67             2922                  LD          H,A              ;[HL] = FRAME_TABLE_PTR^
   0938 E5             2923                  PUSH        HL
   0939 DD6E02         2924                  LD          L,[THIS_SPRITE+STATUS]
   093C DD6603         2925                  LD          H,[THIS_SPRITE+STATUS+1]
   093F 110000         2926                  LD          DE,FRAME
   0942 19             2927                  ADD         HL,DE            ;[HL] = FRAME
   0943 7E             2928                  LD          A,[HL]           ;CALCULATE OFFSET OF
   0944 CB27           2929                  SLA         A                ;COLOR ENTRY
   0946 010000         2930                  LD          BC,0
   0949 4F             2931                  LD          C,A
   094A E1             2932                  POP         HL
   094B 09             2933                  ADD         HL,BC            ;[HL] = COLOR
   094C 7E             2934                  LD          A,[HL]           ;OR IN 80H
   094D F680           2935                  OR          80H
   094F FD7703         2936                  LD          [SPRITE_PTR+COLOR_AND_TAG],A
                       2937
   0952 C30A00         2938                  JP          PUT_Y_AND_NAME
                       2939 *        ELSE
                       2940 ********* CONTINUE BELOW
                       2941
                       2942
                       2943 * BEGIN PUT1SPRITE
          <0955>       2944 PUT1SPRITE       EQU         $
                       2945
                       2946 *    SPRITE_PTR := WORK_BUFFER
   0955 FD2A8006       2947                  LD          SPRITE_PTR,[WORK_BUFFER]
                       2948
                       2949 *    WITH THIS_SPRITE^,SPRITE_PTR^ DO
                       2950
                       2951 *      IF (STATUS^.X_LOCATION > -32) AND (STATUS^.X_LOCATION < 256) AND
                       2952 *         (STATUS^.Y_LOCATION > -32) AND (STATUS^.Y_LOCATION < 192) THEN
   0959 DD6E02         2953                  LD          L,[THIS_SPRITE+STATUS]
   095C DD6603         2954                  LD          H,[THIS_SPRITE+STATUS+1]
   095F 110001         2955                  LD          DE,X_LOCATION
   0962 19             2956                  ADD         HL,DE            ; [HL] = X_LOCATION
   0963 4E             2957                  LD          C,[HL]
   0964 23             2958                  INC         HL
   0965 46             2959                  LD          B,[HL]           ; BC = X_LOCATION
   0966 78             2960                  LD          A,B              ; COMPARE BC WITH -32
   0967 FE00           2961                  CP          0
   0969 2808           2962                  JR          Z,OK__3
   096B FEFF           2963                  CP          -1
   096D C20A54         2964                  JP          NZ,DONT_PUT
   0970 79             2965                  LD          A,C
   0971 FEE1           2966                  CP          -31
   0973 FA0A54         2967                  JP          M,DONT_PUT
   0976                2968 OK__3
   0976 23             2969                  INC         HL               ; [HL] = Y_LOCATION
   0977 4E             2970                  LD          C,[HL]
   0978 23             2971                  INC         HL
   0979 46             2972                  LD          B,[HL]           ; BC = Y_LOCATION
   097A 78             2973                  LD          A,B              ; COMPARE BC WITH -32
   097B FE00           2974                  CP          0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  097D 280B          2975                JR          Z,OK__4
  097F FEFF          2976                CP          -1
  0981 C20A54        2977                JP          NZ,DONT_PUT
  0984 79            2978                LD          A,C
  0985 FEE1          2979                CP          -31
  0987 FA0A54        2980                JP          M,DONT_PUT
  098A               2981 OK__4
                     2982
                     2983 *        IF STATUS^.X_LOCATION < 0 THEN
  098A 2B            2984                DEC         HL
  098B 2B            2985                DEC         HL              ; [HL] = HI(X_LOCATION)
  098C 7E            2986                LD          A,[HL]          ; COMPARE WITH 0
  098D FE00          2987                CP          0
  098F CA09CA        2988                JP          Z,CONTINUE
                     2989
                     2990 *      X := BYTE(STATUS^.X_LOCATION) + 32
  0992 2B            2991                DEC         HL              ;[HL] = ^X_LOCATION
  0993 4E            2992                LD          C,[HL]
  0994 23            2993                INC         HL
  0995 46            2994                LD          B,[HL]
  0996 210020        2995                LD          HL,32
  0999 09            2996                ADD         HL,BC
  099A 7D            2997                LD          A,L
  099B FD7701        2998                LD          [SPRITE_PTR+X],A
                     2999
                     3000 *      COLOR_AND_TAG := GRAPHICS^.FRAME_TABLE[STATUS^.FRAME].COLOR OR 80H
  099E DD6E00        3001                LD          L,[THIS_SPRITE+GRAPHICS]
  09A1 DD6601        3002                LD          H,[THIS_SPRITE+GRAPHICS+1]
  09A4 110005        3003                LD          DE,FRAME_TABLE_PTR
  09A7 19            3004                ADD         HL,DE           ;[HL] = FRAME_TABLE_PTR
  09A8 EB            3005                EX          DE,HL
  09A9 1A            3006                LD          A,[DE]
  09AA 6F            3007                LD          L,A
  09AB 13            3008                INC         DE
  09AC 1A            3009                LD          A,[DE]
  09AD 67            3010                LD          H,A             ;[HL] = FRAME_TABLE_PTR^
  09AE E5            3011                PUSH        HL
  09AF DD6E02        3012                LD          L,[THIS_SPRITE+STATUS]
  09B2 DD6603        3013                LD          H,[THIS_SPRITE+STATUS+1]
  09B5 110000        3014                LD          DE,FRAME
  09B8 19            3015                ADD         HL,DE           ;[HL] = FRAME
  09B9 7E            3016                LD          A,[HL]          ;CALCULATE OFFSET OF
  09BA CB27          3017                SLA         A               ;COLOR ENTRY
  09BC 010000        3018                LD          BC,0
  09BF 4F            3019                LD          C,A
  09C0 E1            3020                POP         HL
  09C1 09            3021                ADD         HL,BC           ;[HL] = COLOR
  09C2 7E            3022                LD          A,[HL]          ;OR IN 80H
  09C3 F680          3023                OR          80H
  09C5 FD7703        3024                LD          [SPRITE_PTR+COLOR_AND_TAG],A
                     3025
  09C8 1836          3026                JR          PUT_Y_AND_NAME
                     3027 *        ELSE
                     3028 ********** CONTINUE FROM HERE
  09CA               3029 CONTINUE
                     3030
                     30''
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     3032
                     3033 *         X := BYTE(STATUS^.X_LOCATION)
  09CA DD6E02        3034                 LD          L,[THIS_SPRITE+STATUS]
  09CD DD6603        3035                 LD          H,[THIS_SPRITE+STATUS+1]
  09D0 110001        3036                 LD          DE,X_LOCATION
  09D3 19            3037                 ADD         HL,DE           ; [HL] = X_LOCATION
  09D4 7E            3038                 LD          A,[HL]
  09D5 FD7701        3039                 LD          [SPRITE_PTR+X],A
                     3040
                     3041 *         COLOR_AND_TAG := GRAPHICS^.FRAME_TABLE[STATUS^.FRAME].COLOR
  09D8 DD6E00        3042                 LD          L,[THIS_SPRITE+GRAPHICS]
  09DB DD6601        3043                 LD          H,[THIS_SPRITE+GRAPHICS+1]
  09DE 110005        3044                 LD          DE,FRAME_TABLE_PTR
  09E1 19            3045                 ADD         HL,DE           ;[HL] = FRAME_TABLE_PTR
  09E2 EB            3046                 EX          DE,HL
  09E3 1A            3047                 LD          A,[DE]
  09E4 6F            3048                 LD          L,A
  09E5 13            3049                 INC         DE
  09E6 1A            3050                 LD          A,[DE]
  09E7 67            3051                 LD          H,A             ;[HL] = FRAME_TABLE_PTR^
  09E8 E5            3052                 PUSH        HL
  09E9 DD6E02        3053                 LD          L,[THIS_SPRITE+STATUS]
  09EC DD6603        3054                 LD          H,[THIS_SPRITE+STATUS+1]
  09EF 110000        3055                 LD          DE,FRAME
  09F2 19            3056                 ADD         HL,DE           ;[HL] = FRAME
  09F3 7E            3057                 LD          A,[HL]          ;CALCULATE OFFSET OF
  09F4 CB27          3058                 SLA         A               ;COLOR ENTRY
  09F6 010000        3059                 LD          BC,0
  09F9 4F            3060                 LD          C,A
  09FA E1            3061                 POP         HL
  09FB 09            3062                 ADD         HL,BC           ;[HL] = COLOR
  09FC 7E            3063                 LD          A,[HL]
  09FD FD7703        3064                 LD          [SPRITE_PTR+COLOR_AND_TAG],A
                     3065
                     3066 *       END_IF
  0A00               3067 PUT_Y_AND_NAME
                     3068
                     3069 *       Y := BYTE(STATUS^.Y_LOCATION)
  0A00 DD6E02        3070                 LD          L,[THIS_SPRITE+STATUS]
  0A03 DD6603        3071                 LD          H,[THIS_SPRITE+STATUS+1]
  0A06 110003        3072                 LD          DE,Y_LOCATION
  0A09 19            3073                 ADD         HL,DE           ; [HL] = Y_LOCATION
  0A0A 7E            3074                 LD          A,[HL]
  0A0B FD7700        3075                 LD          [SPRITE_PTR+Y],A
                     3076
                     3077 *       NAME := GRAPHICS^.FRAME_TABLE[STATUS^.FRAME].SHAPE
                     3078 *               + GRAPHICS^.FIRST_GEN_NAME
  0A0E DD6E00        3079                 LD          L,[THIS_SPRITE+GRAPHICS]
  0A11 DD6601        3080                 LD          H,[THIS_SPRITE+GRAPHICS+1]
  0A14 110005        3081                 LD          DE,FRAME_TABLE_PTR
  0A17 19            3082                 ADD         HL,DE           ;[HL] = FRAME_TABLE_PTR
  0A18 EB            3083                 EX          DE,HL
  0A19 1A            3084                 LD          A,[DE]
  0A1A 6F            3085                 LD          L,A
  0A1B 13            3086                 INC         DE
  0A1C 1A            3087                 LD          A,[DE]
  0A1D 67            3088                 LD          H,A             ;[HL] = FRAME_TABLE_PTR^
```

```
LOCATION OBJECT CODE LINE      SOURCE LINE

  0A1E E5                             PUSH       HL
  0A1F DD6E02                         LD         L,[THIS_SPRITE+STATUS]
  0A22 DD6603                         LD         H,[THIS_SPRITE+STATUS+1]
  0A25 110000                         LD         DE,FRAME
  0A28 19                             ADD        HL,DE            ;[HL] = FRAME
  0A29 7E                             LD         A,[HL]           ;CALCULATE OFFSET OF
  0A2A CB27                           SLA        A                ;SHAPE ENTRY
  0A2C 010000                         LD         BC,0
  0A2F 4F                             LD         C,A
  0A30 E1                             POP        HL
  0A31 09                             ADD        HL,BC
  0A32 23                             INC        HL               ;[HL] = SHAPE
  0A33 7E                             LD         A,[HL]
  0A34 DD6E00                         LD         L,[THIS_SPRITE+GRAPHICS]
  0A37 DD6601                         LD         H,[THIS_SPRITE+GRAPHICS+1]
  0A3A 110001                         LD         DE,FIRST_GEN_NAME
  0A3D 19                             ADD        HL,DE            ;[HL] = FIRST_GEN_NAME
  0A3E 86                             ADD        A,[HL]
  0A3F FD7702                         LD         [SPRITE_PTR+NAME],A

                     *      PUT_VRAM  (0,THIS_SPRITE^.SPRITE_INDEX,SPRITE_PTR,1)
  0A42 AF                             XOR        A
  0A43 1600                           LD         D,0
  0A45 DD5E04                         LD         E,[THIS_SPRITE+SPRITE_INDEX]
  0A48 FDE5                           PUSH       SPRITE_PTR
  0A4A E1                             POP        HL
  0A4B FD210001                       LD         IY,1             ;COUNT OF ONE ITEM
  0A4F CD1FBE                         CALL       PUT_VRAM

  0A52 1832                           JR         EXIT_PUT_SPR
                     *     ELSE
  0A54                   DONT_PUT  ; PUT SPRITE OFF THE SCREEN BY SETTING ITS X AND EARLY CLOCK

                     *      GET_VRAM  (0,THIS_SPRITE^.SPRITE_INDEX,SPRITE_PTR,1)
  0A54 FDE5                           PUSH       SPRITE_PTR       ; SAVE INDEX REGS.
  0A56 DDE5                           PUSH       THIS_SPRITE
  0A58 FDE5                           PUSH       SPRITE_PTR
  0A5A FDE5                           PUSH       SPRITE_PTR
  0A5C AF                             XOR        A
  0A5D 1600                           LD         D,0
  0A5F DD5E04                         LD         E,[THIS_SPRITE+SPRITE_INDEX]
  0A62 E1                             POP        HL
  0A63 FD210001                       LD         IY,1             ;COUNT OF ONE ITEM
  0A67 CD1FBB                         CALL       GET_VRAM

                     *      SPRITE_PTR.X := 0
  0A6A 3E00                           LD         A,0
  0A6C FDE1                           POP        SPRITE_PTR
  0A6E FD7701                         LD         [SPRITE_PTR+X],A

                     *      SPRITE_PTR.COLOR_AND_TAG := 80H
  0A71 3E80                           LD         A,80H
  0A73 FD7703                         LD         [SPRITE_PTR+COLOR_AND_TAG],A

                     *      PUT_VRAM  (0,THIS_SPRITE^.SPRITE_INDEX,SPRITE_PTR,1)
  0A76 AF                             XOR        A
  0A77 1600                           LD
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   0A79 DDE1         3146                 POP            THIS_SPRITE
   0A7B DD5E04       3147                 LD             E,[THIS_SPRITE+SPRITE_INDEX]
   0A7E E1           3148                 POP            HL
   0A7F FD210001     3149                 LD             IY,1            ;COUNT OF ONE ITEM
   0A83 CD1FBE       3150                 CALL           PUT_VRAM
                     3151
                     3152 *      END_IF
                     3153
                     3154 * END PUT0SPRITE,PUT1SPRITE
   0A86              3155 EXIT_PUT_SPR
   0A86 C9           3156                 RET
                     3157  PROG
```

```
LOCATION OBJECT CODE LINE      SOURCE LINE

                   3159 ;
                   3160 ;*********** MODIFIED VERSION TO RUN ON HP ASSEMBLER **************************
                   3161 ;
                   3162 ;                                                              4/16/82
                   3163 ;                                                              13:50:00
                   3164 ;****************** PUT_MOBILE *********************************************
                   3165 ;
                   3166 ;DESCRIPTION:     THIS PROCEDURE PLACES A MOBILE OBJECT ON THE PATTERN PLANE
                   3167 ;                  AT THE X,Y PIXEL LOCATION SPECIFIED IN THAT OBJECT'S RAM STATUS
                   3168 ;                  AREA_
                   3169 ;
                   3170 ;                  A BUFFER AREA OF 204 BYTES (GRAPHICS MODE II) OR 141 BYTES
                   3171 ;                  (GRAPHICS MODE I) IS REQUIRED FOR FORMING THE NEW GENERATORS
                   3172 ;                  REPRESENTING THE OBJECT ON IT'S BACKGROUND_  THE PROCEDURE
                   3173 ;                  USES RAM STARTING AT (F_BUF_SPACE) FOR THIS BUFFER
                   3174 ;
                   3175 ;INPUT:           IX = ADDRESS OF OBJECT TO BE PROCESSED
                   3176 ;                  HL = ADDRESS OF OBJECT'S GRAPHICS TABLES IN ROM
                   3177 ;                   B = SELECTOR FOR METHOD OF COMBINING OBJECT GENERATORS
                   3178 ;                       WITH BACKGROUND GENERATORS
                   3179 ;
                   3180 ;                       1 = OBJECT PATTERN GENS ORed WITH BACKGROUND PATTERN GENS
                   3181 ;                           COLOR1 OF BACKGROUND CHANGED TO MOBILE OBJECT'S COLOR
                   3182 ;                           IF CORRESPONDING PATTERN BYTE NOT ZERO
                   3183 ;
                   3184 ;                       2 = REPLACE BACKGROUND PATTERN GENS WITH OBJECT PATTERN GENS
                   3185 ;                           TREAT COLOR SAME AS #1
                   3186 ;
                   3187 ;                       3 = SAME AS #1 EXCEPT COLOR0 CHANGED TO TRANSPARENT
                   3188 ;
                   3189 ;                       4 = SAME AS #2 EXCEPT COLOR0 CHANGED TO TRANSPARENT
                   3190 ;
                   3191 ;****************************************************************************
                   3192 ;
                   3193 ;)
                   3194 ;          EXT     READ_VRAM,WRITE_VRAM,WORK_BUFFER,GET_VRAM,PUT_VRAM
                   3195 ;          EXT     PX_TO_PTRN_POS,GET_BKGRND,VDP_MODE_WORD,PUTFRAME
                   3196            GLB     PUT_MOBILE
                   3197 ; THE FOLLOWING ARE OFFSETS FROM THE START OF THE FREE BUFFER AREA
                   3198 ; THESE LOCATIONS USED TO STORE VARIABLES AND PATTERN AND COLOR DATA
          <0000>   3199 YDISP    EQU     0        ;Y DISPLACEMENT
          <0001>   3200 XDISP    EQU     1        ;X DISPLACEMENT
          <0002>   3201 COLR     EQU     2        ;COLOR
          <0003>   3202 FLAGS    EQU     3        ;BITS 0,1 = SELECTOR #, BIT X = GRAPHICS MODE [I/II]
          <0004>   3203 FRM      EQU     4        ;FRM   TO BE DISPLAYED
          <0005>   3204 F_GEN    EQU     5        ;NAME OF FIRST GENERATOR IN OBJECT'S GEN TABLE
          <0007>   3205 YP_OS    EQU     7        ;Y_PAT_POS OF OLD_SCREEN
          <0006>   3206 XP_OS    EQU     6        ;X_PAT_POS OF OLD_SCREEN
          <0012>   3207 YP_BK    EQU     18       ;Y_PAT_POS OF BACKGROUND
          <0011>   3208 XP_BK    EQU     17       ;X_PAT_POS OF BACKGROUND
          <001C>   3209 BK_PTN   EQU     28       ;START OF BACKGROUND PATTERN GENERATORS
          <0064>   3210 OBJ_PTN  EQU     100      ;START OF OBJECT'S PATTERN GENERATORS
          <0084>   3211 BK_CLR   EQU     132      ;START OF BACKGROUND COLOR GENERATORS
0A87               3212 PUT_MOBILE
                   3213 ; GET X AND Y LOCATIONS, CONVERT TO X AND Y PATTERN POSITIONS AND X AND Y
                   3214 ; DISPLACEMENTS [AMOUNT BY WHICH OBJECT SHIFTED OFF PATTERN POSITION BOUNDARY]
0A87      8006       15            LD      ORK_I    1          I ST/        FRE       ER A
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  0A88 3A73C3      3216          LD A,[VDP_MODE_WORD]      ;FIND OUT WHICH GRAPHICS MODE WE ARE IN
  0A8B CB4F        3217          BIT 1,A
                   3218 ;        IF [PSW,IS,ZERO]          ;THEN MODE I
  0A90 2004        3219          JR NZ,ELSE1
  0A92 CB88        3220            RES 7,B
  0A94 1802        3221          JR END1
                   3222 ;        ELSE                      ;MODE II
  0A96 CBF8        3223 ELSE1      SET 7,B
                   3224 ;        ENDIF
  0A98 FD7003      3225 END1     LD [IY+FLAGS],B           ;SAVE SELECTOR
  0A9B E5          3226          PUSH HL                   ;SAVE GRAPHICS ADDRESS
  0A9C DD6603      3227          LD H,[IX+3]               ;HL := ADDR_ OF STATUS
  0A9F DD6E02      3228          LD L,[IX+2]
  0AA2 7E          3229          LD A,[HL]                 ;GET FRAME #
  0AA3 FD7704      3230          LD [IY+FRM],A           ;AND SAVE
  0AA6 EE80        3231          XOR 80H                   ;COMPLEMENT TABLE_IN_USE FLAG
  0AA8 77          3232          LD [HL],A                 ;SAVE BACK IN STATUS AREA
  0AA9 23          3233          INC HL                    ;POINT TO X_LOCATION
  0AAA 5E          3234          LD E,[HL]                 ;E := LOW X_LOCATION
  0AAB 7B          3235          LD A,E
  0AAC E607        3236          AND 7                     ;A := #PIXELS TO RIGHT OF PATTERN BOUNDARY
  0AAE ED44        3237          NEG
  0AB0 C608        3238          ADD A,8                   ;AMOUNT TO SHIFT PATTERN LEFT FROM NEXT PAT BOUNDARY
  0AB2 FD7701      3239          LD [IY+XDISP],A           ;SAVE
  0AB5 23          3240          INC HL
  0AB6 56          3241          LD D,[HL]                 ;DE := X_LOCATION
  0AB7 CD07E8      3242          CALL PX_TO_PTRN_POS       ;CALCULATE X_PAT_POS OF BACKGROUND
  0ABA FD7311      3243          LD [IY+XP_BK],E           ;AND SAVE
  0ABD 23          3244          INC HL                    ;POINT TO Y_LOCATION
  0ABE 5E          3245          LD E,[HL]                 ;E := LOW Y_LOCATION
  0ABF 7B          3246          LD A,E
  0AC0 E607        3247          AND 7                     ;A := #PIXELS TO RIGHT OF PATTERN BOUNDARY
  0AC2 FD7700      3248          LD [IY+YDISP],A           ;SAVE
  0AC5 23          3249          INC HL
  0AC6 56          3250          LD D,[HL]                 ;DE := Y_LOCATION
  0AC7 CD07E8      3251          CALL PX_TO_PTRN_POS       ;CALCULATE Y_PAT_POS
  0ACA FD7312      3252          LD [IY+YP_BK],E
                   3253
                   3254 ; NOW GET THE NINE NAMES THAT CONSTITUTE THE BACKGROUND ON WHICH THE MOBILE OBJECT
                   3255 ; WILL BE SUPERIMPOSED
  0ACD 2A8006      3256 PM1      LD HL,[WORK_BUFFER]
  0AD0 110013      3257          LD DE,YP_BK+1             ;POINT TO SPACE FOR BACKGROUND NAMES
  0AD3 19          3258          ADD HL,DE
  0AD4 FD5612      3259          LD D,[IY+YP_BK]           ;D := Y_PAT_POS
  0AD7 FD5E11      3260          LD E,[IY+XP_BK]           ;E := X_PAT_POS
  0ADA 010303      3261          LD BC,303H                ;B := Y_EXTENT, C := X_EXTENT
  0ADD CD0898      3262          CALL GET_BKGRND           ;GET BACKGROUND NAMES
                   3263 ; READ OLD_SCREEN INTO BUFFER AND GET COLOR AND FIRST_GEN_NAME
  0AE0 DD5605      3264 PM2      LD D,[IX+5]               ;DE := OLD_SCREEN ADDRESS
  0AE3 DD5E04      3265          LD E,[IX+4]
  0AE6 DD7E06      3266          LD A,[IX+6]               ;GET FIRST GEN NAME
  0AE9 DDE1        3267          POP IX                    ;IX := ADDRESS OF GRAPHICS
  0AEB FD2A8006    3268          LD IY,[WORK_BUFFER]
  0AEF FD7705      3269          LD [IY+F_GEN],A           ;SAVE IN BUFFER
  0AF2 D5          3270          PUSH DE                   ;SAVE OLD_SCREEN ADDRESS
  0AF3 2A8006      3271          LD HL,[WORK_BUFFER]       ;HL := ADDR OF START OF BUFFER
  0AF6 010006      3272          LD BC,XP_OS               ;SPACE TO MOVE OLD_SCREEN TO
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  0AF9 09            3273           ADD HL,BC
  0AFA 010008        3274           LD BC,11                ;GET 9 NAMES FROM VRAM
                     3275 ;         IF [_D,LT,70H]          ;THEN OLD_SCREEN IS IN VRAM
  0AFD 7A            3276           LD A,D
  0AFE FE70          3277           CP 70H
  0B00 3005          3278           JR NC,ELSE2
  0B02 CD1FE2        3279             CALL READ_VRAM
  0B05 1803          3280             JR END2
                     3281 ;         ELSE                    ;OLD_SCREEN IN CPU RAM
  0B07 EB            3282 ELSE2       EX DE,HL
  0B08 EDB0          3283             LDIR
  0B0A               3284 END2  ;ENDIF
                     3285
                     3286 ; AT THIS POINT, IX = GRAPHICS, [SP] = OLD_SCREEN
                     3287 ; BACKGROUND PATTERN POSITION AND NAMES STARTING AT YP_BK
                     3288 ; OLD_SCREEN PATTERN POSITION AND NAMES STARTING AT YP_OS
                     3289 ; FIND ALL NAMES IN BACKGROUND WHICH BELONG TO THIS OBJECT'S PATTERN GENERATORS
                     3290 ; AND REPLACE WITH NAME FROM OLD_SCREEN WHICH CORRESPONDS TO THAT PATTERN POSITION
  0B0A 2A8006        3291 PM3       LD HL,[WORK_BUFFER]     ;HL := BUFFER BASE
  0B0D 110013        3292           LD DE,YP_BK+1
  0B10 19            3293           ADD HL,DE               ;POINT TO FIRST OF BACKGROUND NAMES
  0B11 D9            3294           EXX
  0B12 ED5B8006      3295           LD DE,[WORK_BUFFER]     ;DE' := BUFFER BASE
  0B16 210008        3296           LD HL,YP_OS+1
  0B19 19            3297           ADD HL,DE
  0B1A EB            3298           EX DE,HL                ;POINTS TO FIRST OF OLD_SCREEN NAMES
  0B1B D9            3299           EXX
  0B1C FD2A8006      3300           LD IY,[WORK_BUFFER]
  0B20 FD4E05        3301           LD C,[IY+F_GEN]         ;C := FIRST_GEN_NAME
                     3302 ;         DO B,9
  0B23 0609          3303           LD B,9
  0B25 7E            3304 DLP1        LD A,[HL]              ;GET A NAME
  0B26 91            3305             SUB C                  ;SUBTRACT FIRST_GEN_NAME
                     3306 ;           IF [_A,LT,18]          ;THEN NAME FALLS IN RANGE OF NAMES FOR OBJECT
  0B27 FE12          3307             CP 18
  0B29 300E          3308             JR NC,END3
                     3309 ;             IF [_A,GE,9]           ;THEN SUB 9 TO FIND CORRECT
  0B2B FE09          3310               CP 9
  0B2D 3802          3311               JR C,END4
  0B2F D609          3312                 SUB 9                ;   POSITION IN OLD_SCREEN
  0B31               3313 END4          ;ENDIF
  0B31 D9            3314               EXX
  0B32 6F            3315               LD L,A               ;FORM A POINTER INTO OLD_SCREEN NAMES
  0B33 2600          3316               LD H,0
  0B35 19            3317               ADD HL,DE
  0B36 7E            3318               LD A,[HL]            ;GET OLD_SCREEN NAME
  0B37 D9            3319               EXX
  0B38 77            3320               LD [HL],A            ;REPLACE BACKGROUND NAME WITH OLD_SCREEN NAME
  0B39               3321 END3        ;ENDIF
  0B39 23            3322           INC HL                  ;POINT TO NEXT NAME IN BACKGROUND
                     3323 ;         ENDDO
  0B3A 10E9          3324           DJNZ DLP1
                     3325
                     3326 ; NOW NEW VERSION OF BACKGROUND NAMES WILL NOT CONTAIN ANY NAMES OF THIS OBJECT
                     3327 ; REPLACE PREVIOUS VERSION OF OLD_SCREEN WITH THIS NEW BACKGROUND
  0B3C D1            3328 PM4       POP DE                  ;DE := OLD SCREEN ADDRESS
  0B3D 2A8006        3329           LD HL,[WORK BUFFER]     ;HL := BUFFER BASE
```