

LOCATION	OBJECT CODE	LINE	SOURCE LINE
0738 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	
0744		2270 ;	IF [A,GE,70H] ;THEN OLD_SCREEN IN CPU RAM
0744 67		2271 EQUAL_TO	
0745 D06E04		2272	LD H,A
0748 7E		2273	LD L,[IX+4]
		2274	LD A,[HL] ;HL := OLD_SCREEN ADDRESS
		2275	
0749 1835		2276	JR END_IF_1
074B		2277 ELSE_1	
		2278	
074B 2A8006		2279 ;	ELSE ;OLD_SCREEN IN VRAM
074E D05605		2280	LD HL,[WORK_BUFFER]
0751 D05E04		2281	LD D,[IX+5] ;GET ADDRESS OF FREE BUFFER SPACE
0754 E5		2282	LD E,[IX+4] ;DE := OLD_SCREEN ADDRESS
0755 D5		2283	PUSH HL ;SAVE 2 COPIES FREE BUFFER ADDR
0756 E5		2284	PUSH DE ;SAVE OLD_SCREEN ADDR
0757 010004		2285	LD BC,4 ;READ 4 BYTES [X,Y,PAT_POSS, X,Y_EXTENTMS]
075A CD103E		2286	CALL VRAM_READ
075E 7E		2287	POP HL ;HL := FREE BUFFER ADDR
075F FE80		2288	CP B0H
0761 2003		2289	JR NZ,GET_OLD
0763 D1		2290	POP DE
0764 1819		2291	JR SKIP_OLD
0766 23		2292	INC HL
0767 23		2293	INC HL
0768 46		2294	LD B,[HL] ;B := X_EXTENT OF OLD_SCREEN
0769 23		2295	INC HL
076A 5E		2296	LD E,[HL] ;E := Y_EXTENT
076B 1600		2297	LD D,0
076D 23		2298	INC HL
076E EB		2299	EX DE,HL ;MULTIPLY X_EXTENT*Y_EXTENT IN HL
		2300	
076F 1801		2301	JR M,XY+1
0771 29		2302	ADD HL,HL
0772 10FD		2303	DJNZ M,XY
		2304	
0774 E5		2305	DJNZ M,XY
0775 C1		2306	PUSH HL
0776 EB		2307 ;	BC := NUMBER OF BYTES TO READ
0777 D1		2308	POP BC ;HL := FREE BUFF ADDR + 4
0778 13		2309	EX DE,HL ;DE := OLD_SCREEN ADDR.
0779 13		2310	POP DE
077A 13		2311	INC DE
077B 13		2312	INC DE
077C CD103E		2313	INC DE
077E E1		2314	INC DE
		2315	CALL VRAM_READ
		2316	POP HL ;READ SAVED NAMES FOR BACKGROUND
		2317 SKIP_OLD	HL := FREE BUFF ADDR.
		2318 ;	ENDIF

LOCATION	OBJECT CODE	LINE	SOURCE LINE
0780		2319	END_IF_1
0780 7E		2320	LD A, [HL]
		2321	; A := X_PAT_POS
		2322	
		2323	
0781 FE80		2324	80H
0783 280F		2325	Z, END_IF_2
		2326	CP
		2327	JR
0785 5E		2328	IF [A, ME, 80H]
0786 23		2329	LD E, [HL]
0787 56		2330	INC HL
0788 23		2331	LD D, [HL]
0789 4E		2332	INC HL
078A 23		2333	LD C, [HL]
078B 46		2334	INC HL
078C 23		2335	LD B, [HL]
		2336	INC HL
		2337	
078D D0E5		2338	PUSH
078F CD0808		2339	CALL PUTFRAME
		2340	; RESTORE OLD_SCREEN TO DISPLAY
0792 D0E1		2341	
		2342	POP
0794		2343	IX ; RESTORE OBJECT POINTER
		2344	END_IF_2
		2345	
0794 E1		2346	ENDIF
0795 D1		2347	POP HL
0796 C1		2348	POP DE
0797 C5		2349	POP BC
0798 D5		2350	PUSH BC
0799 E5		2351	PUSH DE
079A D06605		2352	PUSH HL
079D D06E04		2353	LD H, [IX+5]
		2354	LD L, [IX+4]
		2355	
07A0 3E70		2356	LD
07A2 BC		2357	CP
07A3 3A03		2358	JR
07A5 2A0006		2359	
07A8		2360	IF [H, LT, 70H]
		2361	LD HL, [WORK_BUFFER]
		2362	
		2363	END_IF_3
		2364	
07A8 73		2365	ENDIF
07A9 23		2366	LD [HL], E
07AA 72		2367	INC HL
07AB 23		2368	LD [HL], D
07AC 71		2369	INC HL
07AD 23		2370	LD [HL], C
07AE 70		2371	INC HL
07AF 23		2372	LD [HL], B
		2373	INC HL
		2374	
07A		2375	INTI

LOCATION	OBJECT CODE	LINE	SOURCE LINE
07B2	CD0898	2376	CALL GET_BKGRND
07B5	D0E1	2377	IX ;RESTORE OBJECT POINTER
07B7	E1	2378	POP HL ;WHERE NAMES ARE IN CPU RAM
07B8	D1	2379	POP DE ;WHERE TO MOVE THEM TO IN VRAM (NAME TABLE)
07B9	C1	2380	POP BC ;HOW MANY TO MOVE
07BA	D0E5	2381	PUSH ;SAVE OBJECT POINTER
07BC	CD0808	2382	CALL PUTFRAME
07BF	D0E1	2383	IX ;RESTORE OBJECT POINTER
07C1	D05605	2384	LD D, [IX+5] ;SEE IF SAVED BACKGROUND TO BE MOVED TO VRAM
07C4	3E70	2385	LD A, 70H
07C6	BA	2386	CP D
07C7	281E	2387	JR Z, END_IF_4
07C9	381C	2388	JR C, END_IF_4
07CB	D05E04	2389	LD E, [IX+4] ;DE := OLD SCREEN ADDR
07CE	D9	2390	EXX ;USE *REG FOR CALCULATION
07CF	2A0006	2391	LD HL, [WORK_BUFFER] ;WHERE NEXT OLD_SCREEN DATA IS
07D2	E5	2392	PUSH HL
07D3	23	2393	INC HL
07D4	23	2394	INC HL
07D5	5E	2395	LD E, [HL]
07D6	1600	2396	LD D, 0
07D8	23	2397	INC HL
07D9	46	2398	LD B, [HL] ;B := Y_EXTENT
07DA	EB	2399	EX DE, HL ;HL := X_EXTENT
07DB	1801	2400	JR M, XY2+1 ;HL := X_EXTENT*Y_EXTENT
07DE	10FD	2401	ADD HL, HL
07E0	E5	2402	PUSH HL
07E1	D9	2403	EXX
07E2	C1	2404	POP BC
07E3	E1	2405	POP HL
07E4	CD1D01	2406	CALL VRAM_WRITE
07E7		2407	END_IF_4
07E7	C9	2408	RET
<p>***** PX.10.PTRN.POS *****</p> <p>2424 DIVIDES REG DE BY B, IF SIGNED RESULT > 127 THEN E := MAX SIGNED</p> <p>2425 POSITIVE NUMBER. IF RESULT < -128, THEN E := MIN NEGATIVE NUM</p> <p>2426 DE = 16 BIT SIGNED NUMBER</p> <p>2427 INPUT: DE/B < -128 E = -128</p> <p>2428 OUTPUT: -128 <= DE/B <=+127 E = DE/B</p> <p>2429 +127 < DE/B E = +127</p> <p>2430 *****</p> <p>2431 2432 ;</p>			

```

LOCATION OBJECT CODE LINE SOURCE LINE
07E8 2433 PX_TO_PTRN_POS GLB PX_TO_PTRN_POS
07E8 E5 2434 PX_TO_PTRN_POS
07E9 CB2A 2435 PX_TO_PTRN_POS
07E8 CB1B 2436 PUSH HL
07ED CB2A 2437 SRA D
07EF CB1B 2438 RR E
07F1 CB2A 2439 SRA D
07F3 CB1B 2440 RR E
07F5 CB7A 2441 SRA D
07F7 2009 2442 RR E
07F9 21FF00 2443 BIT 7,D
07FC 19 2444 JR NZ,NEGIV
07FD E1 2445 LD HL,OFF80H
07FE D0 2446 ADD HL,DE
07FF 1E7F 2447 POP HL
0801 C9 2448 RET MC
2449 LD E,7FH
2450 RET
2451 ;XXXXX
2452 ;XXXXX LD HL,080H
2453 NEGIV: 2454 ;XXXXX
2455 ADD HL,DE
2456 POP HL
2457 RET C
2458 LD E,80H
2459 RET
2460 ;***** PUT FRAME *****
2461 ;***** THE NAMES WHICH CONSTITUTE A FRAME ARE MOVED TO THE NAME TABLE *****
2462 ;***** IN VRAM. THE UPPER LEFT HAND CORNER OF THE FRAME IS POSITIONED *****
2463 ;***** AT X_PAT_POS, Y_PAT_POS. *****
2464 ;***** INPUT: *****
2465 ;***** 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 ;***** PUTFRAME *****
2473 ;***** COPY PARAMETERS INTO PRIMED REGISTERS *****
2474 ;***** AND FRAME ADDRESS INTO DE *****
2475 ;*****
2476 ;*****
2477 ;*****
2478 ;*****
2479 ;*****
2480 ;*****
2481 ;*****
2482 ;*****
2483 ;*****
2484 ;*****
2485 PF1: LD A,E
2486 BIT 7,A
2487 JR NZ,XP_NEG
2488 CP 32
2489 RET I
2490 ;*****
2491 ;*****
2492 ;*****
2493 ;*****
2494 ;*****
2495 ;*****
2496 ;*****
2497 ;*****
2498 ;*****
2499 ;*****
2500 ;*****
2501 ;*****
2502 ;*****
2503 ;*****
2504 ;*****
2505 ;*****
2506 ;*****
2507 ;*****
2508 ;*****
2509 ;*****
2510 ;*****
2511 ;*****
2512 ;*****
2513 ;*****
2514 ;*****
2515 ;*****
2516 ;*****
2517 ;*****
2518 ;*****
2519 ;*****
2520 ;*****
2521 ;*****
2522 ;*****
2523 ;*****
2524 ;*****
2525 ;*****
2526 ;*****
2527 ;*****
2528 ;*****
2529 ;*****
2530 ;*****
2531 ;*****
2532 ;*****
2533 ;*****
2534 ;*****
2535 ;*****
2536 ;*****
2537 ;*****
2538 ;*****
2539 ;*****
2540 ;*****
2541 ;*****
2542 ;*****
2543 ;*****
2544 ;*****
2545 ;*****
2546 ;*****
2547 ;*****
2548 ;*****
2549 ;*****
2550 ;*****
2551 ;*****
2552 ;*****
2553 ;*****
2554 ;*****
2555 ;*****
2556 ;*****
2557 ;*****
2558 ;*****
2559 ;*****
2560 ;*****
2561 ;*****
2562 ;*****
2563 ;*****
2564 ;*****
2565 ;*****
2566 ;*****
2567 ;*****
2568 ;*****
2569 ;*****
2570 ;*****
2571 ;*****
2572 ;*****
2573 ;*****
2574 ;*****
2575 ;*****
2576 ;*****
2577 ;*****
2578 ;*****
2579 ;*****
2580 ;*****
2581 ;*****
2582 ;*****
2583 ;*****
2584 ;*****
2585 ;*****
2586 ;*****
2587 ;*****
2588 ;*****
2589 ;*****
2590 ;*****
2591 ;*****
2592 ;*****
2593 ;*****
2594 ;*****
2595 ;*****
2596 ;*****
2597 ;*****
2598 ;*****
2599 ;*****
2600 ;*****
2601 ;*****
2602 ;*****
2603 ;*****
2604 ;*****
2605 ;*****
2606 ;*****
2607 ;*****
2608 ;*****
2609 ;*****
2610 ;*****
2611 ;*****
2612 ;*****
2613 ;*****
2614 ;*****
2615 ;*****
2616 ;*****
2617 ;*****
2618 ;*****
2619 ;*****
2620 ;*****
2621 ;*****
2622 ;*****
2623 ;*****
2624 ;*****
2625 ;*****
2626 ;*****
2627 ;*****
2628 ;*****
2629 ;*****
2630 ;*****
2631 ;*****
2632 ;*****
2633 ;*****
2634 ;*****
2635 ;*****
2636 ;*****
2637 ;*****
2638 ;*****
2639 ;*****
2640 ;*****
2641 ;*****
2642 ;*****
2643 ;*****
2644 ;*****
2645 ;*****
2646 ;*****
2647 ;*****
2648 ;*****
2649 ;*****
2650 ;*****
2651 ;*****
2652 ;*****
2653 ;*****
2654 ;*****
2655 ;*****
2656 ;*****
2657 ;*****
2658 ;*****
2659 ;*****
2660 ;*****
2661 ;*****
2662 ;*****
2663 ;*****
2664 ;*****
2665 ;*****
2666 ;*****
2667 ;*****
2668 ;*****
2669 ;*****
2670 ;*****
2671 ;*****
2672 ;*****
2673 ;*****
2674 ;*****
2675 ;*****
2676 ;*****
2677 ;*****
2678 ;*****
2679 ;*****
2680 ;*****
2681 ;*****
2682 ;*****
2683 ;*****
2684 ;*****
2685 ;*****
2686 ;*****
2687 ;*****
2688 ;*****
2689 ;*****
2690 ;*****
2691 ;*****
2692 ;*****
2693 ;*****
2694 ;*****
2695 ;*****
2696 ;*****
2697 ;*****
2698 ;*****
2699 ;*****
2700 ;*****
2701 ;*****
2702 ;*****
2703 ;*****
2704 ;*****
2705 ;*****
2706 ;*****
2707 ;*****
2708 ;*****
2709 ;*****
2710 ;*****
2711 ;*****
2712 ;*****
2713 ;*****
2714 ;*****
2715 ;*****
2716 ;*****
2717 ;*****
2718 ;*****
2719 ;*****
2720 ;*****
2721 ;*****
2722 ;*****
2723 ;*****
2724 ;*****
2725 ;*****
2726 ;*****
2727 ;*****
2728 ;*****
2729 ;*****
2730 ;*****
2731 ;*****
2732 ;*****
2733 ;*****
2734 ;*****
2735 ;*****
2736 ;*****
2737 ;*****
2738 ;*****
2739 ;*****
2740 ;*****
2741 ;*****
2742 ;*****
2743 ;*****
2744 ;*****
2745 ;*****
2746 ;*****
2747 ;*****
2748 ;*****
2749 ;*****
2750 ;*****
2751 ;*****
2752 ;*****
2753 ;*****
2754 ;*****
2755 ;*****
2756 ;*****
2757 ;*****
2758 ;*****
2759 ;*****
2760 ;*****
2761 ;*****
2762 ;*****
2763 ;*****
2764 ;*****
2765 ;*****
2766 ;*****
2767 ;*****
2768 ;*****
2769 ;*****
2770 ;*****
2771 ;*****
2772 ;*****
2773 ;*****
2774 ;*****
2775 ;*****
2776 ;*****
2777 ;*****
2778 ;*****
2779 ;*****
2780 ;*****
2781 ;*****
2782 ;*****
2783 ;*****
2784 ;*****
2785 ;*****
2786 ;*****
2787 ;*****
2788 ;*****
2789 ;*****
2790 ;*****
2791 ;*****
2792 ;*****
2793 ;*****
2794 ;*****
2795 ;*****
2796 ;*****
2797 ;*****
2798 ;*****
2799 ;*****
2800 ;*****
2801 ;*****
2802 ;*****
2803 ;*****
2804 ;*****
2805 ;*****
2806 ;*****
2807 ;*****
2808 ;*****
2809 ;*****
2810 ;*****
2811 ;*****
2812 ;*****
2813 ;*****
2814 ;*****
2815 ;*****
2816 ;*****
2817 ;*****
2818 ;*****
2819 ;*****
2820 ;*****
2821 ;*****
2822 ;*****
2823 ;*****
2824 ;*****
2825 ;*****
2826 ;*****
2827 ;*****
2828 ;*****
2829 ;*****
2830 ;*****
2831 ;*****
2832 ;*****
2833 ;*****
2834 ;*****
2835 ;*****
2836 ;*****
2837 ;*****
2838 ;*****
2839 ;*****
2840 ;*****
2841 ;*****
2842 ;*****
2843 ;*****
2844 ;*****
2845 ;*****
2846 ;*****
2847 ;*****
2848 ;*****
2849 ;*****
2850 ;*****
2851 ;*****
2852 ;*****
2853 ;*****
2854 ;*****
2855 ;*****
2856 ;*****
2857 ;*****
2858 ;*****
2859 ;*****
2860 ;*****
2861 ;*****
2862 ;*****
2863 ;*****
2864 ;*****
2865 ;*****
2866 ;*****
2867 ;*****
2868 ;*****
2869 ;*****
2870 ;*****
2871 ;*****
2872 ;*****
2873 ;*****
2874 ;*****
2875 ;*****
2876 ;*****
2877 ;*****
2878 ;*****
2879 ;*****
2880 ;*****
2881 ;*****
2882 ;*****
2883 ;*****
2884 ;*****
2885 ;*****
2886 ;*****
2887 ;*****
2888 ;*****
2889 ;*****
2890 ;*****
2891 ;*****
2892 ;*****
2893 ;*****
2894 ;*****
2895 ;*****
2896 ;*****
2897 ;*****
2898 ;*****
2899 ;*****
2900 ;*****
2901 ;*****
2902 ;*****
2903 ;*****
2904 ;*****
2905 ;*****
2906 ;*****
2907 ;*****
2908 ;*****
2909 ;*****
2910 ;*****
2911 ;*****
2912 ;*****
2913 ;*****
2914 ;*****
2915 ;*****
2916 ;*****
2917 ;*****
2918 ;*****
2919 ;*****
2920 ;*****
2921 ;*****
2922 ;*****
2923 ;*****
2924 ;*****
2925 ;*****
2926 ;*****
2927 ;*****
2928 ;*****
2929 ;*****
2930 ;*****
2931 ;*****
2932 ;*****
2933 ;*****
2934 ;*****
2935 ;*****
2936 ;*****
2937 ;*****
2938 ;*****
2939 ;*****
2940 ;*****
2941 ;*****
2942 ;*****
2943 ;*****
2944 ;*****
2945 ;*****
2946 ;*****
2947 ;*****
2948 ;*****
2949 ;*****
2950 ;*****
2951 ;*****
2952 ;*****
2953 ;*****
2954 ;*****
2955 ;*****
2956 ;*****
2957 ;*****
2958 ;*****
2959 ;*****
2960 ;*****
2961 ;*****
2962 ;*****
2963 ;*****
2964 ;*****
2965 ;*****
2966 ;*****
2967 ;*****
2968 ;*****
2969 ;*****
2970 ;*****
2971 ;*****
2972 ;*****
2973 ;*****
2974 ;*****
2975 ;*****
2976 ;*****
2977 ;*****
2978 ;*****
2979 ;*****
2980 ;*****
2981 ;*****
2982 ;*****
2983 ;*****
2984 ;*****
2985 ;*****
2986 ;*****
2987 ;*****
2988 ;*****
2989 ;*****
2990 ;*****
2991 ;*****
2992 ;*****
2993 ;*****
2994 ;*****
2995 ;*****
2996 ;*****
2997 ;*****
2998 ;*****
2999 ;*****
3000 ;*****

```

LOCATION	OBJECT CODE	LINE	SOURCE LINE
081E 81	2490	XP_NEG: ADD A,C	; A := X_PAT_POS + X_EXTENT
081F C87F	2491	BIT 7,A	; IS A NEG?
0821 C0	2492	RET NZ	; YES, RETURN
0822 87	2493	OR A	; A = 07
0823 C0	2494	RET Z	; RETURN IF 0
	2495	;XXXXXX	
	2496	;X.IN.BOUNDS::	
0824	2497		
	2498	X_IN_BOUNDS	
	2499		
	2500	; IF [L,E,IS,MIMUS]	; IF X_PAT_POS < 0 , FRAME BLEEDING ON FROM LEFT
0824 C878	2501		7,E
0826 2820	2502	BIT JR	Z,ELSE_0
	2503		
	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	;XXXXXX	
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:	; NUMBER OF NAMES ON SCREEN EACH ROW
	2513	;XXXXXX	
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 78	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 E8	2526	EX DE,HL	
0842 09	2527	ADD HL,BC	; ADD DISPLACEMENT TO NAME TABLE POINTER
0843 E8	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	
0848 78	2537		
0849 81	2538	PF2: LD A,E	; IS X_PAT_POS + X_EXTENT > 31
	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

OCCATION	OBJECT CODE LINE	SOURCE LINE
0002 D9	2604	EXX
0003 D1	2605	POP DE
0004 C1	2606	POP BC
	2607 ;	ENDIF
	2608 ;	ENDIF
0005	2609	ENDIF
	2610	END_IF_10
	2611	
0005 D9	2612	EXX
0006 C5	2613	PUSH BC
0007 0600	2614	LD B,0
0009 D9	2615	ADD HL,BC
000A E8	2616	EX DE,HL
000B 010020	2617	LD BC,32
000E D9	2618	ADD HL,BC
000F E8	2619	EX DE,HL
0090 C1	2620	POP BC
0091 D9	2621	EXX
0092 1C	2622	INC E
	2623 ;	UNTIL (.E, EQ, .B)
	2624	
0093 7B	2625	LD A,E
0094 B8	2626	CP B
0095 20CF	2627	JR NZ,RPT_1
0097 C9	2628	
	2629	RET
	2630 ;	
	2631 ;	COMMENT)
	2632	***** GET_BKGRND *****
	2633	THIS ROUTINE GETS THE NAMES FROM THE NAME TABLE WHICH CONSTITUTE
	2634	THE BACKGROUND ON WHICH AN OBJECT IS TO BE MOVED
	2635	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
0098	2643	GET_BKGRND:
0098 CD08C0	2644	CALL CALC_OFFSET
0098 C5	2645	PUSH BC
009C 0600	2646	LD B,0
009E C5	2647	PUSH BC
009F FDE1	2648	POP IY
00A1 C1	2649	POP BC
	2650 ;	REPEAT
	2651	
00A2	2652	RPT_2
	2653	
00A2 C5	2654	PUSH BC
00A3 D5	2655	PUSH DE
00A4 E5	2656	PUSH HL
00A5 FDE5	2657	PUSH IY
00A7 3E02	2658	LD A,2
00A9 CD1BA3	2659	CALL GET_VRAM
00AC FDE1	2660	POP IY

; INCREMENT POINTER INTO FRAME BY X_EXTENT

; INCREMENT OFFSET BY 32

; UNTIL Y=Y_EXTENT

***** GET_BKGRND *****
 THIS ROUTINE GETS THE NAMES FROM THE NAME TABLE WHICH CONSTITUTE
 THE BACKGROUND ON WHICH AN OBJECT IS TO BE MOVED
 HL = LOCATION IN CPU RAM TO WHICH THE NAMES ARE MOVED
 D = Y_PAT_POS (TOP ROW OF PATTERN)
 E = X_PAT_POS (LEFT HAND COLUMN)
 B = Y_EXTENT OF PATTERN
 C = X_EXTENT OF PATTERN

GLB GET_BKGRND
 ; OFFSET INTO NAME TABLE OF POSITION OF UPPER LEFT
 ; HAND PATTERN
 ; GET X_EXTENT INTO IY
 ; NUMBER OF NAMES PER ROW
 ; Y_EXTENT-1 TIMES

; TABLE CODE FOR PATTERN NAME TABLE
 LD A,2
 CALL GET_VRAM
 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
2761 *   END SPRITE_OBJECT
2762
2763 * SPRITE_GRAPHICS = RECORD
2764 *   OBJECT_TYPE: BYTE
2765 *   FIRST_GEN_NAME: BYTE
2766 *   PTRN_POINTER: ^PATTERN_GENERATOR
2767 *   MUMGEN: BYTE
2768 *   FRAME_TABLE_PTR: ^ARRAY[0..10] OF FRAME (TABLE OF ANIMATION FRAMES)
2769 *   END SPRITE_ROM_GRAPHICS
2770
2771 * SPRITE_STATUS = RECORD
2772 *   FRAME: BYTE
2773 *   X_LOCATION: INTEGER
2774 *   Y_LOCATION: INTEGER
2775 *   NEXT_GEN: BYTE
2776 *   END SPRITE_STATUS
2777
2778 * FRAME = RECORD
2779 *   COLOR: BYTE
2780 *   SHAPE: BYTE
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 ***** DICTIONARY *****
2792
2793
2794
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
2798 SPRITE_PTR EQU
2799 * SPRITE_PTR IS A POINTER TO THE NEW SPRITE NAME TABLE ENTRY BEING
2800 * BUILT BY THIS ROUTINE.
2801
2802
2803 * THIS SPRITE IS A POINTER TO THE SPRITE OBJECT IN THE ROM

```

<000F>

<0


```

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 MUNGEN        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_VRAM (TABLE CODE:BYTE; START_INDEX:SLICE:BYTE;
2836   VAR DATA:BUFFER;ITEM_COUNT:INTEGER);
2837
2838 * EXTERNAL PROCEDURE GET_VRAM (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 LY
2849
2850 ***** PROCEDURE BODY *****
2851
2852      PROG
2853      GLB PUTOSPRITE,PUT1SPRITE
2854
2855 * BEGIN PUTOSPRITE
2856 PUTOSPRITE      EQU      5
2857
2858 * SPRITE_PTR := WORK_BUFFER
2859      LD
2860
080F FD2A8006      SPRITE_PTR,[WORK_BUFFER]

```

LOCATION OBJECT CODE LINE SOURCE LINE

```

2861 * WITH THIS_SPRITE^,SPRITE_PTR^ DO
2862
2863 * IF (STATUS^.X_LOCATION > -B) AND (STATUS^.X_LOCATION < 256) AND
2864 * (STATUS^.Y_LOCATION > -B) AND (STATUS^.Y_LOCATION < 192) THEN
2865 * LD
2866 * LD
2867 * LD
2868 * ADD
2869 * LD
2870 * INC
2871 * LD
2872 * LD
2873 * CP
2874 * JR
2875 * CP
2876 * JP
2877 * LD
2878 * CP
2879 * JP
2880 * OK_1
2881 * INC
2882 * LD
2883 * INC
2884 * LD
2885 * LD
2886 * CP
2887 * JR
2888 * CP
2889 * JP
2890 * LD
2891 * CP
2892 * JP
2893 * OK_2
2894 *
2895 * IF STATUS^.X_LOCATION < 0 THEN
2896 * DEC
2897 * DEC
2898 * LD
2899 * CP
2900 * JP
2901 *
2902 * X := BYTE(STATUS^.X_LOCATION) + B
2903 * DEC
2904 * LD
2905 * INC
2906 * LD
2907 * LD
2908 * ADD
2909 * LD
2910 * LD
2911 *
2912 * COLOR_AND_TAG := GRAPHICS^.FRAME_TABLE[STATUS^.FRAME].COLOR OR B0H
2913 * LD
2914 * LD
2915 * LD
2916 * ADD
2917 *
08E3 D06E02
08E6 D06603
08E9 110001
08EC 19
08ED 4E
08EE 23
08EF 46
08F0 7B
08F1 FE00
08F3 2B08
08F5 FEFF
08F7 C20A54
08FA 79
08FB FEF9
08FD FA0A54
0900
0900 23
0901 4E
0902 23
0903 46
0904 7B
0905 FE00
0907 2B08
0909 FEFF
090B C20A54
090E 79
090F FEF9
0911 FA0A54
0914
0914 2B
0915 2B
0916 7E
0917 FE00
0919 CA09CA
091C 2B
091D 4E
091E 23
091F 46
0920 210008
0923 09
0924 7D
0925 FD7701
0928 D06E00
092B D06601
092E 110005
0931 19
0932

```

```

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
0938 E5 2923 PUSH HL
0939 D06E02 2924 LD L,[THIS_SPRITE+STATUS]
093C D06603 2925 H,[THIS_SPRITE+STATUS+1]
093F 110000 2926 DE,FRAME
0942 19 2927 LD HL,DE
0943 7E 2928 LD A,[HL]
0944 CB27 2929 ADD A
0946 010000 2930 BC,0
0949 4F 2931 LD C,A
094A E1 2932 POP HL
094B 09 2933 HL,BC
094C 7E 2934 LD A,[HL]
094D F680 2935 OR DON
094F FD7703 2936 LD ISPRITE_PTR,COLOR_AND_TAG,A
0952 C30A00 2937 PUT_Y_AND_NAME
0953 * 2938 JP
0954 * 2939 * ELSE
0955 * 2940 ***** CONTINUE BELOW
0956 * 2941
0957 * 2942
0958 * 2943 * BEGIN PUT1SPRITE
0959 * 2944 PUT1SPRITE EQU $
0960 * 2945
0961 * 2946 * SPRITE_PTR := WORK_BUFFER
0962 * 2947 LD SPRITE_PTR,WORK_BUFFER
0963 * 2948
0964 * 2949 * WITH THIS_SPRITE^,SPRITE_PTR^ DO
0965 * 2950
0966 * 2951 * IF (STATUS^X_LOCATION > -32) AND (STATUS^X_LOCATION < 256) AND
0967 * 2952 * (STATUS^Y_LOCATION > -32) AND (STATUS^Y_LOCATION < 192) THEN
0968 * 2953 L,[THIS_SPRITE+STATUS]
0969 * 2954 DE,X_LOCATION
0970 * 2955 HL,DE
0971 * 2956 C,[HL]
0972 * 2957 HL
0973 * 2958 B,[HL]
0974 * 2959 A,B
0975 * 2960 0
0976 * 2961 Z,OK_3
0977 * 2962 -1
0978 * 2963 NZ,DONT_PUT
0979 * 2964 A,C
0980 * 2965 -31
0981 * 2966 M,DONT_PUT
0982 * 2967 JP
0983 * 2968 OK_3
0984 * 2969 INC
0985 * 2970 LD
0986 * 2971 INC
0987 * 2972 LD
0988 * 2973 A,B
0989 * 2974 0
0990 * 2975
0991 * 2976
0992 * 2977
0993 * 2978
0994 * 2979
0995 * 2980
0996 * 2981
0997 * 2982
0998 * 2983
0999 * 2984

```

```

LOCATION    OBJECT CODE LINE    SOURCE LINE
097D 2808      2975      JR
097F FEFF      2976      CP
0981 C20A54    2977      JP
0984 79        2978      LD
0985 FEE1      2979      CP
0987 FA0A54    2980      JP
098A           2981      OK_4
0982           2982
098A 2B        2983      *
098B 2B        2984      DEC HL
098C 7E        2985      DEC HL
098D FE00      2986      LD A,[HL]
098F CA09CA    2987      CP
0988           2988      JP
0989           2989
0990 *         2990      X := BYTE(STATUS^.X_LOCATION) + 32
0992 2B        2991      DEC HL
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
0999 *         2999
099E D06E00    3000      *
09A1 D06601    3001      COLOR_AND_TAG := GRAPHICS^.FRAME_TABLE[STATUS^.FRAME].COLOR OR 80H
09A4 110005    3002      LD L,[THIS_SPRITE+GRAPHICS]
09A7 19        3003      LD L,[THIS_SPRITE+GRAPHICS+1]
09A8 E8        3004      ADD DE,FRAME_TABLE_PTR
09A9 1A        3005      EX DE,HL
09AA 6F        3006      LD A,[DE]
09AB 13        3007      LD L,A
09AC 7E        3008      INC DE
09AD 67        3009      LD A,[DE]
09AE E5        3010      LD H,A
09AF D06E02    3011      PUSH HL
09B2 D06603    3012      LD L,[THIS_SPRITE+STATUS]
09B5 110000    3013      LD L,[THIS_SPRITE+STATUS+1]
09B8 19        3014      LD DE,FRAME
09B9 7E        3015      ADD HL,DE
09BA CB27      3016      LD A,[HL]
09BC 010000    3017      SLA A
09BF 4F        3018      LD BC,0
09C0 E1        3019      LD C,A
09C1 09        3020      POP HL
09C2 7E        3021      ADD HL,BC
09C3 F680      3022      LD A,[HL]
09C5 FD7703    3023      OR 80H
09C8 1836      3024      LD [SPRITE_PTR+COLOR_AND_TAG],A
09C9           3025
09C8 1836      3026      JR
09C9           3027      *
09C9           3028      ELSE
09C9           3029      ***** CONTINUE FROM HERE
09C9           3030      CONTINUE
09C9           3031      Y0..

```

LOCATION OBJECT CODE LINE SOURCE LINE

```

3032 *
3033 * X := BYTE(STATUS^.X_LOCATION)
3034 LD L,[THIS_SPRITE+STATUS]
3035 LD H,[THIS_SPRITE+STATUS+1]
3036 LD DE,X_LOCATION
3037 ADD HL,DE ; [HL] = X_LOCATION
3038 LD A,[HL]
3039 LD [SPRITE_PTR+X],A
3040
3041 * COLOR_AND_TAG := GRAPHICS^.FRAME_TABLE[STATUS^.FRAME].COLOR
3042 LD L,[THIS_SPRITE+GRAPHICS]
3043 LD H,[THIS_SPRITE+GRAPHICS+1]
3044 LD DE,FRAME_TABLE_PTR
3045 ADD HL,DE ; [HL] = FRAME_TABLE_PTR
3046 EX DE,HL
3047 LD A,[DE]
3048 LD L,A
3049 INC DE
3050 LD A,[DE]
3051 LD H,A ; [HL] = FRAME_TABLE_PTR^
3052 PUSH HL
3053 LD L,[THIS_SPRITE+STATUS]
3054 LD H,[THIS_SPRITE+STATUS+1]
3055 LD DE,FRAME^
3056 ADD HL,DE ; [HL] = FRAME
3057 LD A,[HL] ; CALCULATE OFFSET OF
3058 SLA A ; COLOR ENTRY
3059 LD BC,0
3060 LD C,A
3061 HL,BC
3062 ADD HL,A ; [HL] = COLOR
3063 LD A,[HL]
3064 LD [SPRITE_PTR+COLOR_AND_TAG],A
3065
3066 * END IF
3067 PUT_Y_AND_NAME
3068
3069 * Y := BYTE(STATUS^.Y_LOCATION)
3070 LD L,[THIS_SPRITE+STATUS]
3071 LD H,[THIS_SPRITE+STATUS+1]
3072 LD DE,Y_LOCATION
3073 ADD HL,DE ; [HL] = Y_LOCATION
3074 LD A,[HL]
3075 LD [SPRITE_PTR+Y],A
3076
3077 * NAME := GRAPHICS^.FRAME_TABLE[STATUS^.FRAME].SHAPE
3078 * + GRAPHICS^.FIRST_GEN_NAME
3079 LD L,[THIS_SPRITE+GRAPHICS]
3080 LD H,[THIS_SPRITE+GRAPHICS+1]
3081 LD DE,FRAME_TABLE_PTR
3082 ADD HL,DE ; [HL] = FRAME_TABLE_PTR
3083 EX DE,HL
3084 LD A,[DE]
3085 LD L,A
3086 INC DE
3087 LD A,[DE]
3088 LD H,A ; [HL] = FRAME_TABLE_PTR^

```

```

LOCATION OBJECT CODE LINE SOURCE LINE
0A1E E5 3089 HL
0A1F D06E02 3090 PUSH
0A22 D06603 3091 LD
0A25 110000 3092 LD
0A28 19 3093 LD
0A29 7E 3094 LD
0A2A C827 3095 SLA
0A2C 010000 3096 LD
0A2F 4F 3097 LD
0A30 E1 3098 POP
0A31 09 3099 ADD
0A32 23 3100 TMC
0A33 7E 3101 LD
0A34 D06E00 3102 LD
0A37 D06601 3103 LD
0A3A 110001 3104 LD
0A3D 19 3105 ADD
0A3E 06 3106 ADD
0A3F FD7702 3107 LD
3108
3109 * PUT_VRAM (0,THIS_SPRITE^.SPRITE_INDEX,SPRITE_PTR,1)
3110 XOR A
3111 LD D,0
3112 LD E,[THIS_SPRITE+SPRITE_INDEX]
3113 PUSH SPRITE_PTR
3114 POP HL
3115 LD IX,1
3116 CALL PUT_VRAM
3117
3118 JR EXIT_PUT_SPR
3119 * ELSE
3120 DONT_PUT ; PUT SPRITE OFF THE SCREEN BY SETTING ITS X AND EARLY CLOCK
3121
3122 * GET_VRAM (0,THIS_SPRITE^.SPRITE_INDEX,SPRITE_PTR,1)
3123 PUSH SPRITE_PTR
3124 PUSH THIS_SPRITE
3125 PUSH SPRITE_PTR
3126 PUSH SPRITE_PTR
3127 XOR A
3128 LD D,0
3129 LD E,[THIS_SPRITE+SPRITE_INDEX]
3130 POP HL
3131 LD IX,1
3132 CALL GET_VRAM
3133
3134 * SPRITE_PTR.X := 0
3135 LD A,0
3136 POP SPRITE_PTR
3137 LD [SPRITE_PTR+X],A
3138
3139 * SPRITE_PTR.COLOR AND TAG := 80H
3140 LD A,80H
3141 LD [SPRITE_PTR+COLOR AND TAG],A
3142
3143 * PUT_VRAM (0,THIS_SPRITE^.SPRITE_INDEX,SPRITE_PTR,1)
3144 XOR A
3145 LD

```

LOCATION	OBJECT	CODE	LINE	SOURCE	LINE
0A79	D0E1		3146	POP	THIS_SPRITE
0A7B	D05E04		3147	LD	E,[THIS_SPRITE+SPRITE_INDEX]
0A7E	E1		3148	POP	HL
0A7F	FD210001		3149	LD	IV,1
0A83	CD17#E		3150	CALL	PUT_VRAM
			3151		
			3152	* END_IF	
			3153		
			3154	* END_PUTOSPRITE,PUTISPRITE	
			3155	EXIT_PUT_SPR	
0A86			3156	RET	
0A86	C9		3157	PROG	

THIS_SPRITE
E,[THIS_SPRITE+SPRITE_INDEX]
HL
IV,1
;COUNT OF ONE ITEM
PUT_VRAM

LOCATION OBJECT CODE LINE SOURCE LINE

```

3159 ; ***** MODIFIED VERSION TO RUN ON HP ASSEMBLER *****
3160 ;
3161 ;
3162 ;
3163 ;
3164 ; ***** PUT MOBILE *****
3165 ;
3166 ; DESCRIPTION:
3167 ; THIS PROCEDURE PLACES A MOBILE OBJECT ON THE PATTERN PLANE
3168 ; AT THE X,Y PIXEL LOCATION SPECIFIED IN THAT OBJECT'S RAM STATUS
3169 ; AREA
3170 ;
3171 ; A BUFFER AREA OF 204 BYTES (GRAPHICS MODE 11) OR 141 BYTES
3172 ; (GRAPHICS MODE 1) IS REQUIRED FOR FORMING THE NEW GENERATORS
3173 ; REPRESENTING THE OBJECT ON IT'S BACKGROUND. THE PROCEDURE
3174 ; USES RAM STARTING AT (F_BUF_SPACE) FOR THIS BUFFER
3175 ;
3176 ; INPUT:
3177 ; HL = ADDRESS OF OBJECT'S GRAPHICS TABLES IN ROM
3178 ; B = SELECTOR FOR METHOD OF COMBINING OBJECT GENERATORS
3179 ; WITH BACKGROUND GENERATORS
3180 ;
3181 ; 1 = OBJECT PATTERN GENS ORED WITH BACKGROUND PATTERN GENS
3182 ; COLOR1 OF BACKGROUND CHANGED TO MOBILE OBJECT'S COLOR
3183 ; IF CORRESPONDING PATTERN BYTE NOT ZERO
3184 ;
3185 ; 2 = REPLACE BACKGROUND PATTERN GENS WITH OBJECT PATTERN GENS
3186 ; TREAT COLOR SAME AS #1
3187 ;
3188 ; 3 = SAME AS #1 EXCEPT COLOR0 CHANGED TO TRANSPARENT
3189 ;
3190 ; 4 = SAME AS #2 EXCEPT COLOR0 CHANGED TO TRANSPARENT
3191 ;
3192 ; *****
3193 ;
3194 ;
3195 ;
3196 ;
3197 ; EXT READ VRAH,WRITE VRAH,WORK_BUFFER,GET VRAH,PUT VRAH
3198 ; PX TO PTRN_POS,GET_BKGRND,VDP_MODE_WORD,PUTFRAME
3199 ; GL8 PUT MOBILE
3200 ;
3201 ; THE FOLLOWING ARE OFFSETS FROM THE START OF THE FREE BUFFER AREA
3202 ; THESE LOCATIONS USED TO STORE VARIABLES AND PATTERN AND COLOR DATA
3203 ;
3204 ; Y DISPLACEMENT
3205 ; X DISPLACEMENT
3206 ; COLOR
3207 ; BITS 0,1 = SELECTOR #, BIT X = GRAPHICS MODE [1/11]
3208 ; FRM TO BE DISPLAYED
3209 ; NAME OF FIRST GENERATOR IN OBJECT'S GEN TABLE
3210 ; Y_PAT_POS OF OLD SCREEN
3211 ; X_PAT_POS OF OLD SCREEN
3212 ; Y_PAT_POS OF BACKGROUND
3213 ; X_PAT_POS OF BACKGROUND
3214 ; START OF BACKGROUND PATTERN GENERATORS
3215 ; START OF OBJECT'S PATTERN GENERATORS
3216 ; START OF BACKGROUND COLOR GENERATORS
3217 ;
3218 ; PUT MOBILE
3219 ;
3220 ; GET X AND Y LOCATIONS, CONVERT TO X AND Y PATTERN POSITIONS AND X AND Y
3221 ; DISPLACEMENTS (AMOUNT BY WHICH OBJECT SHIFTED OFF PATTERN POSITION BOUNDARY)
3222 ;
3223 ; LD ORK.1 11 T ST/ FRE ER #
3224 ;
3225 ;
3226 ;
3227 ;
3228 ;
3229 ;
3230 ;
3231 ;
3232 ;
3233 ;
3234 ;
3235 ;
3236 ;
3237 ;
3238 ;
3239 ;
3240 ;
3241 ;
3242 ;
3243 ;
3244 ;
3245 ;
3246 ;
3247 ;
3248 ;
3249 ;
3250 ;
3251 ;
3252 ;
3253 ;
3254 ;
3255 ;
3256 ;
3257 ;
3258 ;
3259 ;
3260 ;
3261 ;
3262 ;
3263 ;
3264 ;
3265 ;
3266 ;
3267 ;
3268 ;
3269 ;
3270 ;
3271 ;
3272 ;
3273 ;
3274 ;
3275 ;
3276 ;
3277 ;
3278 ;
3279 ;
3280 ;
3281 ;
3282 ;
3283 ;
3284 ;
3285 ;
3286 ;
3287 ;
3288 ;
3289 ;
3290 ;
3291 ;
3292 ;
3293 ;
3294 ;
3295 ;
3296 ;
3297 ;
3298 ;
3299 ;
3300 ;
3301 ;
3302 ;
3303 ;
3304 ;
3305 ;
3306 ;
3307 ;
3308 ;
3309 ;
3310 ;
3311 ;
3312 ;
3313 ;
3314 ;
3315 ;
3316 ;
3317 ;
3318 ;
3319 ;
3320 ;
3321 ;
3322 ;
3323 ;
3324 ;
3325 ;
3326 ;
3327 ;
3328 ;
3329 ;
3330 ;
3331 ;
3332 ;
3333 ;
3334 ;
3335 ;
3336 ;
3337 ;
3338 ;
3339 ;
3340 ;
3341 ;
3342 ;
3343 ;
3344 ;
3345 ;
3346 ;
3347 ;
3348 ;
3349 ;
3350 ;
3351 ;
3352 ;
3353 ;
3354 ;
3355 ;
3356 ;
3357 ;
3358 ;
3359 ;
3360 ;
3361 ;
3362 ;
3363 ;
3364 ;
3365 ;
3366 ;
3367 ;
3368 ;
3369 ;
3370 ;
3371 ;
3372 ;
3373 ;
3374 ;
3375 ;
3376 ;
3377 ;
3378 ;
3379 ;
3380 ;
3381 ;
3382 ;
3383 ;
3384 ;
3385 ;
3386 ;
3387 ;
3388 ;
3389 ;
3390 ;
3391 ;
3392 ;
3393 ;
3394 ;
3395 ;
3396 ;
3397 ;
3398 ;
3399 ;
3400 ;
3401 ;
3402 ;
3403 ;
3404 ;
3405 ;
3406 ;
3407 ;
3408 ;
3409 ;
3410 ;
3411 ;
3412 ;
3413 ;
3414 ;
3415 ;
3416 ;
3417 ;
3418 ;
3419 ;
3420 ;
3421 ;
3422 ;
3423 ;
3424 ;
3425 ;
3426 ;
3427 ;
3428 ;
3429 ;
3430 ;
3431 ;
3432 ;
3433 ;
3434 ;
3435 ;
3436 ;
3437 ;
3438 ;
3439 ;
3440 ;
3441 ;
3442 ;
3443 ;
3444 ;
3445 ;
3446 ;
3447 ;
3448 ;
3449 ;
3450 ;
3451 ;
3452 ;
3453 ;
3454 ;
3455 ;
3456 ;
3457 ;
3458 ;
3459 ;
3460 ;
3461 ;
3462 ;
3463 ;
3464 ;
3465 ;
3466 ;
3467 ;
3468 ;
3469 ;
3470 ;
3471 ;
3472 ;
3473 ;
3474 ;
3475 ;
3476 ;
3477 ;
3478 ;
3479 ;
3480 ;
3481 ;
3482 ;
3483 ;
3484 ;
3485 ;
3486 ;
3487 ;
3488 ;
3489 ;
3490 ;
3491 ;
3492 ;
3493 ;
3494 ;
3495 ;
3496 ;
3497 ;
3498 ;
3499 ;
3500 ;
3501 ;
3502 ;
3503 ;
3504 ;
3505 ;
3506 ;
3507 ;
3508 ;
3509 ;
3510 ;
3511 ;
3512 ;
3513 ;
3514 ;
3515 ;
3516 ;
3517 ;
3518 ;
3519 ;
3520 ;
3521 ;
3522 ;
3523 ;
3524 ;
3525 ;
3526 ;
3527 ;
3528 ;
3529 ;
3530 ;
3531 ;
3532 ;
3533 ;
3534 ;
3535 ;
3536 ;
3537 ;
3538 ;
3539 ;
3540 ;
3541 ;
3542 ;
3543 ;
3544 ;
3545 ;
3546 ;
3547 ;
3548 ;
3549 ;
3550 ;
3551 ;
3552 ;
3553 ;
3554 ;
3555 ;
3556 ;
3557 ;
3558 ;
3559 ;
3560 ;
3561 ;
3562 ;
3563 ;
3564 ;
3565 ;
3566 ;
3567 ;
3568 ;
3569 ;
3570 ;
3571 ;
3572 ;
3573 ;
3574 ;
3575 ;
3576 ;
3577 ;
3578 ;
3579 ;
3580 ;
3581 ;
3582 ;
3583 ;
3584 ;
3585 ;
3586 ;
3587 ;
3588 ;
3589 ;
3590 ;
3591 ;
3592 ;
3593 ;
3594 ;
3595 ;
3596 ;
3597 ;
3598 ;
3599 ;
3600 ;
3601 ;
3602 ;
3603 ;
3604 ;
3605 ;
3606 ;
3607 ;
3608 ;
3609 ;
3610 ;
3611 ;
3612 ;
3613 ;
3614 ;
3615 ;
3616 ;
3617 ;
3618 ;
3619 ;
3620 ;
3621 ;
3622 ;
3623 ;
3624 ;
3625 ;
3626 ;
3627 ;
3628 ;
3629 ;
3630 ;
3631 ;
3632 ;
3633 ;
3634 ;
3635 ;
3636 ;
3637 ;
3638 ;
3639 ;
3640 ;
3641 ;
3642 ;
3643 ;
3644 ;
3645 ;
3646 ;
3647 ;
3648 ;
3649 ;
3650 ;
3651 ;
3652 ;
3653 ;
3654 ;
3655 ;
3656 ;
3657 ;
3658 ;
3659 ;
3660 ;
3661 ;
3662 ;
3663 ;
3664 ;
3665 ;
3666 ;
3667 ;
3668 ;
3669 ;
3670 ;
3671 ;
3672 ;
3673 ;
3674 ;
3675 ;
3676 ;
3677 ;
3678 ;
3679 ;
3680 ;
3681 ;
3682 ;
3683 ;
3684 ;
3685 ;
3686 ;
3687 ;
3688 ;
3689 ;
3690 ;
3691 ;
3692 ;
3693 ;
3694 ;
3695 ;
3696 ;
3697 ;
3698 ;
3699 ;
3700 ;
3701 ;
3702 ;
3703 ;
3704 ;
3705 ;
3706 ;
3707 ;
3708 ;
3709 ;
3710 ;
3711 ;
3712 ;
3713 ;
3714 ;
3715 ;
3716 ;
3717 ;
3718 ;
3719 ;
3720 ;
3721 ;
3722 ;
3723 ;
3724 ;
3725 ;
3726 ;
3727 ;
3728 ;
3729 ;
3730 ;
3731 ;
3732 ;
3733 ;
3734 ;
3735 ;
3736 ;
3737 ;
3738 ;
3739 ;
3740 ;
3741 ;
3742 ;
3743 ;
3744 ;
3745 ;
3746 ;
3747 ;
3748 ;
3749 ;
3750 ;
3751 ;
3752 ;
3753 ;
3754 ;
3755 ;
3756 ;
3757 ;
3758 ;
3759 ;
3760 ;
3761 ;
3762 ;
3763 ;
3764 ;
3765 ;
3766 ;
3767 ;
3768 ;
3769 ;
3770 ;
3771 ;
3772 ;
3773 ;
3774 ;
3775 ;
3776 ;
3777 ;
3778 ;
3779 ;
3780 ;
3781 ;
3782 ;
3783 ;
3784 ;
3785 ;
3786 ;
3787 ;
3788 ;
3789 ;
3790 ;
3791 ;
3792 ;
3793 ;
3794 ;
3795 ;
3796 ;
3797 ;
3798 ;
3799 ;
3800 ;
3801 ;
3802 ;
3803 ;
3804 ;
3805 ;
3806 ;
3807 ;
3808 ;
3809 ;
3810 ;
3811 ;
3812 ;
3813 ;
3814 ;
3815 ;
3816 ;
3817 ;
3818 ;
3819 ;
3820 ;
3821 ;
3822 ;
3823 ;
3824 ;
3825 ;
3826 ;
3827 ;
3828 ;
3829 ;
3830 ;
3831 ;
3832 ;
3833 ;
3834 ;
3835 ;
3836 ;
3837 ;
3838 ;
3839 ;
3840 ;
3841 ;
3842 ;
3843 ;
3844 ;
3845 ;
3846 ;
3847 ;
3848 ;
3849 ;
3850 ;
3851 ;
3852 ;
3853 ;
3854 ;
3855 ;
3856 ;
3857 ;
3858 ;
3859 ;
3860 ;
3861 ;
3862 ;
3863 ;
3864 ;
3865 ;
3866 ;
3867 ;
3868 ;
3869 ;
3870 ;
3871 ;
3872 ;
3873 ;
3874 ;
3875 ;
3876 ;
3877 ;
3878 ;
3879 ;
3880 ;
3881 ;
3882 ;
3883 ;
3884 ;
3885 ;
3886 ;
3887 ;
3888 ;
3889 ;
3890 ;
3891 ;
3892 ;
3893 ;
3894 ;
3895 ;
3896 ;
3897 ;
3898 ;
3899 ;
3900 ;
3901 ;
3902 ;
3903 ;
3904 ;
3905 ;
3906 ;
3907 ;
3908 ;
3909 ;
3910 ;
3911 ;
3912 ;
3913 ;
3914 ;
3915 ;
3916 ;
3917 ;
3918 ;
3919 ;
3920 ;
3921 ;
3922 ;
3923 ;
3924 ;
3925 ;
3926 ;
3927 ;
3928 ;
3929 ;
3930 ;
3931 ;
3932 ;
3933 ;
3934 ;
3935 ;
3936 ;
3937 ;
3938 ;
3939 ;
3940 ;
3941 ;
3942 ;
3943 ;
3944 ;
3945 ;
3946 ;
3947 ;
3948 ;
3949 ;
3950 ;
3951 ;
3952 ;
3953 ;
3954 ;
3955 ;
3956 ;
3957 ;
3958 ;
3959 ;
3960 ;
3961 ;
3962 ;
3963 ;
3964 ;
3965 ;
3966 ;
3967 ;
3968 ;
3969 ;
3970 ;
3971 ;
3972 ;
3973 ;
3974 ;
3975 ;
3976 ;
3977 ;
3978 ;
3979 ;
3980 ;
3981 ;
3982 ;
3983 ;
3984 ;
3985 ;
3986 ;
3987 ;
3988 ;
3989 ;
3990 ;
3991 ;
3992 ;
3993 ;
3994 ;
3995 ;
3996 ;
3997 ;
3998 ;
3999 ;
4000 ;

```

0A87

0A87

LOCATION OBJECT CODE LINE SOURCE LINE

```

0A08 3A73C3      LD A, [VDP_MODE_WORD]      ; FIND OUT WHICH GRAPHICS MODE WE ARE IN
0A0E CB4F      BIT 1, A
3216          ; THEN MODE 1
3217          JR NZ, ELSE1
0A90 2004      RES 7, B
0A92 CB88      JR END1
0A94 1802      ELSE
3222          ; MODE 11
3223          SET 7, B
3224          ENDF
0A98 FD7003      LD (1Y+FLAGS), B
3225          PUSH HL
0A9C D06603      LD H, (1X+3)
3226          LD L, (1X+2)
0A9F D06E02      LD A, [HL]
3227          XOR B0H
3228          INC HL
0AA2 7E      LD E, [HL]
3229          LD A, [HL]
3230          LD (1Y+FRM), A
0AA3 FD7704      XOR B0H
3231          LD (HL), A
0AA6 EE00      INC HL
3232          LD E, [HL]
0AA8 77      LD A, E
3233          AND 7
3234          NEG
3235          ADD A, B
0AAE ED44      LD (1Y+YDISP), A
3236          INC HL
0AB0 C608      LD D, [HL]
3237          CALL PX TO PTRN_POS
0AB2 FD7701      LD (1Y+XP_BK), E
3238          INC HL
0AB5 23      LD E, [HL]
3239          LD A, E
0AB6 56      AND 7
3240          LD D, [HL]
0AB7 CD07E8      CALL PX TO PTRN_POS
3241          LD (1Y+XP_BK), E
0ABA FD7311      INC HL
3242          LD E, [HL]
0ABD 23      LD A, E
3243          AND 7
0ABE 5E      LD (1Y+YDISP), A
3244          INC HL
0ABF 78      LD D, [HL]
3245          LD A, E
0AC0 E607      AND 7
3246          LD (1Y+YDISP), A
0AC2 FD7700      INC HL
3247          LD D, [HL]
0AC5 23      LD A, E
3248          LD (1Y+XP_BK), E
0AC6 56      CALL PX TO PTRN_POS
3249          LD (1Y+YP_BK), E
0AC7 CD07E8      ; NOW GET THE NINE NAMES THAT CONSTITUTE THE BACKGROUND ON WHICH THE MOBILE OBJECT
0ACA FD7312      ; WILL BE SUPERIMPOSED
3250          LD HL, [WORK_BUFFER]
0ACD 2A8006      LD DE, YP_BK+1
3251          ADD HL, DE
0AD0 110013      LD D, (1Y+YP_BK)
3252          LD E, (1Y+XP_BK)
0AD3 19      LD BC, 303H
3253          CALL GET_BKGRND
0AD4 FD5612      LD D, (1X+5)
3254          LD E, (1X+4)
0AD7 FD5E11      LD A, (1X+6)
3255          POP IX
0ADA 010303      LD 1Y, [WORK_BUFFER]
3256          PUSH DE
0AD0 CD0898      LD HL, (1Y+F_GEN), A
3257          LD BC, XP_OS
3258          LD D, (1X+5)
3259          LD E, (1X+4)
3260          LD A, (1X+6)
0AE0 D05605      POP IX
3261          LD 1Y, [WORK_BUFFER]
0AE3 D05E04      LD HL, (1Y+F_GEN), A
3262          PUSH DE
0AE6 D07E06      LD HL, [WORK_BUFFER]
3263          LD BC, XP_OS
0AE9 D0E1      LD D, (1X+5)
3264          LD E, (1X+4)
0AEB FD2A8006      LD A, (1X+6)
3265          LD 1Y, [WORK_BUFFER]
0AEF FD7705      LD HL, (1Y+F_GEN), A
3266          PUSH DE
0AF2 05      LD BC, XP_OS
3267          LD HL, [WORK_BUFFER]
0AF3 2A8006      LD D, (1X+5)
3268          LD E, (1X+4)
0AF6 010006      LD A, (1X+6)
3269          POP IX
3270          LD 1Y, [WORK_BUFFER]
3271          LD HL, (1Y+F_GEN), A
3272          PUSH DE
3273          LD BC, XP_OS
3274          LD HL, [WORK_BUFFER]
3275          LD D, (1X+5)
3276          LD E, (1X+4)
3277          LD A, (1X+6)
3278          POP IX
3279          LD 1Y, [WORK_BUFFER]
3280          LD HL, (1Y+F_GEN), A
3281          PUSH DE
3282          LD BC, XP_OS
3283          LD HL, [WORK_BUFFER]
3284          LD D, (1X+5)
3285          LD E, (1X+4)
3286          LD A, (1X+6)
3287          POP IX
3288          LD 1Y, [WORK_BUFFER]
3289          LD HL, (1Y+F_GEN), A
3290          PUSH DE
3291          LD BC, XP_OS
3292          LD HL, [WORK_BUFFER]
3293          LD D, (1X+5)
3294          LD E, (1X+4)
3295          LD A, (1X+6)
3296          POP IX
3297          LD 1Y, [WORK_BUFFER]
3298          LD HL, (1Y+F_GEN), A
3299          PUSH DE
3300          LD BC, XP_OS
3301          LD HL, [WORK_BUFFER]
3302          LD D, (1X+5)
3303          LD E, (1X+4)
3304          LD A, (1X+6)
3305          POP IX
3306          LD 1Y, [WORK_BUFFER]
3307          LD HL, (1Y+F_GEN), A
3308          PUSH DE
3309          LD BC, XP_OS
3310          LD HL, [WORK_BUFFER]
3311          LD D, (1X+5)
3312          LD E, (1X+4)
3313          LD A, (1X+6)
3314          POP IX
3315          LD 1Y, [WORK_BUFFER]
3316          LD HL, (1Y+F_GEN), A
3317          PUSH DE
3318          LD BC, XP_OS
3319          LD HL, [WORK_BUFFER]
3320          LD D, (1X+5)
3321          LD E, (1X+4)
3322          LD A, (1X+6)
3323          POP IX
3324          LD 1Y, [WORK_BUFFER]
3325          LD HL, (1Y+F_GEN), A
3326          PUSH DE
3327          LD BC, XP_OS
3328          LD HL, [WORK_BUFFER]
3329          LD D, (1X+5)
3330          LD E, (1X+4)
3331          LD A, (1X+6)
3332          POP IX
3333          LD 1Y, [WORK_BUFFER]
3334          LD HL, (1Y+F_GEN), A
3335          PUSH DE
3336          LD BC, XP_OS
3337          LD HL, [WORK_BUFFER]
3338          LD D, (1X+5)
3339          LD E, (1X+4)
3340          LD A, (1X+6)
3341          POP IX
3342          LD 1Y, [WORK_BUFFER]
3343          LD HL, (1Y+F_GEN), A
3344          PUSH DE
3345          LD BC, XP_OS
3346          LD HL, [WORK_BUFFER]
3347          LD D, (1X+5)
3348          LD E, (1X+4)
3349          LD A, (1X+6)
3350          POP IX
3351          LD 1Y, [WORK_BUFFER]
3352          LD HL, (1Y+F_GEN), A
3353          PUSH DE
3354          LD BC, XP_OS
3355          LD HL, [WORK_BUFFER]
3356          LD D, (1X+5)
3357          LD E, (1X+4)
3358          LD A, (1X+6)
3359          POP IX
3360          LD 1Y, [WORK_BUFFER]
3361          LD HL, (1Y+F_GEN), A
3362          PUSH DE
3363          LD BC, XP_OS
3364          LD HL, [WORK_BUFFER]
3365          LD D, (1X+5)
3366          LD E, (1X+4)
3367          LD A, (1X+6)
3368          POP IX
3369          LD 1Y, [WORK_BUFFER]
3370          LD HL, (1Y+F_GEN), A
3371          PUSH DE
3372          LD BC, XP_OS
3373          LD HL, [WORK_BUFFER]
3374          LD D, (1X+5)
3375          LD E, (1X+4)
3376          LD A, (1X+6)
3377          POP IX
3378          LD 1Y, [WORK_BUFFER]
3379          LD HL, (1Y+F_GEN), A
3380          PUSH DE
3381          LD BC, XP_OS
3382          LD HL, [WORK_BUFFER]
3383          LD D, (1X+5)
3384          LD E, (1X+4)
3385          LD A, (1X+6)
3386          POP IX
3387          LD 1Y, [WORK_BUFFER]
3388          LD HL, (1Y+F_GEN), A
3389          PUSH DE
3390          LD BC, XP_OS
3391          LD HL, [WORK_BUFFER]
3392          LD D, (1X+5)
3393          LD E, (1X+4)
3394          LD A, (1X+6)
3395          POP IX
3396          LD 1Y, [WORK_BUFFER]
3397          LD HL, (1Y+F_GEN), A
3398          PUSH DE
3399          LD BC, XP_OS
3400          LD HL, [WORK_BUFFER]
3401          LD D, (1X+5)
3402          LD E, (1X+4)
3403          LD A, (1X+6)
3404          POP IX
3405          LD 1Y, [WORK_BUFFER]
3406          LD HL, (1Y+F_GEN), A
3407          PUSH DE
3408          LD BC, XP_OS
3409          LD HL, [WORK_BUFFER]
3410          LD D, (1X+5)
3411          LD E, (1X+4)
3412          LD A, (1X+6)
3413          POP IX
3414          LD 1Y, [WORK_BUFFER]
3415          LD HL, (1Y+F_GEN), A
3416          PUSH DE
3417          LD BC, XP_OS
3418          LD HL, [WORK_BUFFER]
3419          LD D, (1X+5)
3420          LD E, (1X+4)
3421          LD A, (1X+6)
3422          POP IX
3423          LD 1Y, [WORK_BUFFER]
3424          LD HL, (1Y+F_GEN), A
3425          PUSH DE
3426          LD BC, XP_OS
3427          LD HL, [WORK_BUFFER]
3428          LD D, (1X+5)
3429          LD E, (1X+4)
3430          LD A, (1X+6)
3431          POP IX
3432          LD 1Y, [WORK_BUFFER]
3433          LD HL, (1Y+F_GEN), A
3434          PUSH DE
3435          LD BC, XP_OS
3436          LD HL, [WORK_BUFFER]
3437          LD D, (1X+5)
3438          LD E, (1X+4)
3439          LD A, (1X+6)
3440          POP IX
3441          LD 1Y, [WORK_BUFFER]
3442          LD HL, (1Y+F_GEN), A
3443          PUSH DE
3444          LD BC, XP_OS
3445          LD HL, [WORK_BUFFER]
3446          LD D, (1X+5)
3447          LD E, (1X+4)
3448          LD A, (1X+6)
3449          POP IX
3450          LD 1Y, [WORK_BUFFER]
3451          LD HL, (1Y+F_GEN), A
3452          PUSH DE
3453          LD BC, XP_OS
3454          LD HL, [WORK_BUFFER]
3455          LD D, (1X+5)
3456          LD E, (1X+4)
3457          LD A, (1X+6)
3458          POP IX
3459          LD 1Y, [WORK_BUFFER]
3460          LD HL, (1Y+F_GEN), A
3461          PUSH DE
3462          LD BC, XP_OS
3463          LD HL, [WORK_BUFFER]
3464          LD D, (1X+5)
3465          LD E, (1X+4)
3466          LD A, (1X+6)
3467          POP IX
3468          LD 1Y, [WORK_BUFFER]
3469          LD HL, (1Y+F_GEN), A
3470          PUSH DE
3471          LD BC, XP_OS
3472          LD HL, [WORK_BUFFER]
3473          LD D, (1X+5)
3474          LD E, (1X+4)
3475          LD A, (1X+6)
3476          POP IX
3477          LD 1Y, [WORK_BUFFER]
3478          LD HL, (1Y+F_GEN), A
3479          PUSH DE
3480          LD BC, XP_OS
3481          LD HL, [WORK_BUFFER]
3482          LD D, (1X+5)
3483          LD E, (1X+4)
3484          LD A, (1X+6)
3485          POP IX
3486          LD 1Y, [WORK_BUFFER]
3487          LD HL, (1Y+F_GEN), A
3488          PUSH DE
3489          LD BC, XP_OS
3490          LD HL, [WORK_BUFFER]
3491          LD D, (1X+5)
3492          LD E, (1X+4)
3493          LD A, (1X+6)
3494          POP IX
3495          LD 1Y, [WORK_BUFFER]
3496          LD HL, (1Y+F_GEN), A
3497          PUSH DE
3498          LD BC, XP_OS
3499          LD HL, [WORK_BUFFER]
3500          LD D, (1X+5)
3501          LD E, (1X+4)
3502          LD A, (1X+6)
3503          POP IX
3504          LD 1Y, [WORK_BUFFER]
3505          LD HL, (1Y+F_GEN), A
3506          PUSH DE
3507          LD BC, XP_OS
3508          LD HL, [WORK_BUFFER]
3509          LD D, (1X+5)
3510          LD E, (1X+4)
3511          LD A, (1X+6)
3512          POP IX
3513          LD 1Y, [WORK_BUFFER]
3514          LD HL, (1Y+F_GEN), A
3515          PUSH DE
3516          LD BC, XP_OS
3517          LD HL, [WORK_BUFFER]
3518          LD D, (1X+5)
3519          LD E, (1X+4)
3520          LD A, (1X+6)
3521          POP IX
3522          LD 1Y, [WORK_BUFFER]
3523          LD HL, (1Y+F_GEN), A
3524          PUSH DE
3525          LD BC, XP_OS
3526          LD HL, [WORK_BUFFER]
3527          LD D, (1X+5)
3528          LD E, (1X+4)
3529          LD A, (1X+6)
3530          POP IX
3531          LD 1Y, [WORK_BUFFER]
3532          LD HL, (1Y+F_GEN), A
3533          PUSH DE
3534          LD BC, XP_OS
3535          LD HL, [WORK_BUFFER]
3536          LD D, (1X+5)
3537          LD E, (1X+4)
3538          LD A, (1X+6)
3539          POP IX
3540          LD 1Y, [WORK_BUFFER]
3541          LD HL, (1Y+F_GEN), A
3542          PUSH DE
3543          LD BC, XP_OS
3544          LD HL, [WORK_BUFFER]
3545          LD D, (1X+5)
3546          LD E, (1X+4)
3547          LD A, (1X+6)
3548          POP IX
3549          LD 1Y, [WORK_BUFFER]
3550          LD HL, (1Y+F_GEN), A
3551          PUSH DE
3552          LD BC, XP_OS
3553          LD HL, [WORK_BUFFER]
3554          LD D, (1X+5)
3555          LD E, (1X+4)
3556          LD A, (1X+6)
3557          POP IX
3558          LD 1Y, [WORK_BUFFER]
3559          LD HL, (1Y+F_GEN), A
3560          PUSH DE
3561          LD BC, XP_OS
3562          LD HL, [WORK_BUFFER]
3563          LD D, (1X+5)
3564          LD E, (1X+4)
3565          LD A, (1X+6)
3566          POP IX
3567          LD 1Y, [WORK_BUFFER]
3568          LD HL, (1Y+F_GEN), A
3569          PUSH DE
3570          LD BC, XP_OS
3571          LD HL, [WORK_BUFFER]
3572          LD D, (1X+5)
3573          LD E, (1X+4)
3574          LD A, (1X+6)
3575          POP IX
3576          LD 1Y, [WORK_BUFFER]
3577          LD HL, (1Y+F_GEN), A
3578          PUSH DE
3579          LD BC, XP_OS
3580          LD HL, [WORK_BUFFER]
3581          LD D, (1X+5)
3582          LD E, (1X+4)
3583          LD A, (1X+6)
3584          POP IX
3585          LD 1Y, [WORK_BUFFER]
3586          LD HL, (1Y+F_GEN), A
3587          PUSH DE
3588          LD BC, XP_OS
3589          LD HL, [WORK_BUFFER]
3590          LD D, (1X+5)
3591          LD E, (1X+4)
3592          LD A, (1X+6)
3593          POP IX
3594          LD 1Y, [WORK_BUFFER]
3595          LD HL, (1Y+F_GEN), A
3596          PUSH DE
3597          LD BC, XP_OS
3598          LD HL, [WORK_BUFFER]
3599          LD D, (1X+5)
3600          LD E, (1X+4)
3601          LD A, (1X+6)
3602          POP IX
3603          LD 1Y, [WORK_BUFFER]
3604          LD HL, (1Y+F_GEN), A
3605          PUSH DE
3606          LD BC, XP_OS
3607          LD HL, [WORK_BUFFER]
3608          LD D, (1X+5)
3609          LD E, (1X+4)
3610          LD A, (1X+6)
3611          POP IX
3612          LD 1Y, [WORK_BUFFER]
3613          LD HL, (1Y+F_GEN), A
3614          PUSH DE
3615          LD BC, XP_OS
3616          LD HL, [WORK_BUFFER]
3617          LD D, (1X+5)
3618          LD E, (1X+4)
3619          LD A, (1X+6)
3620          POP IX
3621          LD 1Y, [WORK_BUFFER]
3622          LD HL, (1Y+F_GEN), A
3623          PUSH DE
3624          LD BC, XP_OS
3625          LD HL, [WORK_BUFFER]
3626          LD D, (1X+5)
3627          LD E, (1X+4)
3628          LD A, (1X+6)
3629          POP IX
3630          LD 1Y, [WORK_BUFFER]
3631          LD HL, (1Y+F_GEN), A
3632          PUSH DE
3633          LD BC, XP_OS
3634          LD HL, [WORK_BUFFER]
3635          LD D, (1X+5)
3636          LD E, (1X+4)
3637          LD A, (1X+6)
3638          POP IX
3639          LD 1Y, [WORK_BUFFER]
3640          LD HL, (1Y+F_GEN), A
3641          PUSH DE
3642          LD BC, XP_OS
3643          LD HL, [WORK_BUFFER]
3644          LD D, (1X+5)
3645          LD E, (1X+4)
3646          LD A, (1X+6)
3647          POP IX
3648          LD 1Y, [WORK_BUFFER]
3649          LD HL, (1Y+F_GEN), A
3650          PUSH DE
3651          LD BC, XP_OS
3652          LD HL, [WORK_BUFFER]
3653          LD D, (1X+5)
3654          LD E, (1X+4)
3655          LD A, (1X+6)
3656          POP IX
3657          LD 1Y, [WORK_BUFFER]
3658          LD HL, (1Y+F_GEN), A
3659          PUSH DE
3660          LD BC, XP_OS
3661          LD HL, [WORK_BUFFER]
3662          LD D, (1X+5)
3663          LD E, (1X+4)
3664          LD A, (1X+6)
3665          POP IX
3666          LD 1Y, [WORK_BUFFER]
3667          LD HL, (1Y+F_GEN), A
3668          PUSH DE
3669          LD BC, XP_OS
3670          LD HL, [WORK_BUFFER]
3671          LD D, (1X+5)
3672          LD E, (1X+4)
3673          LD A, (1X+6)
3674          POP IX
3675          LD 1Y, [WORK_BUFFER]
3676          LD HL, (1Y+F_GEN), A
3677          PUSH DE
3678          LD BC, XP_OS
3679          LD HL, [WORK_BUFFER]
3680          LD D, (1X+5)
3681          LD E, (1X+4)
3682          LD A, (1X+6)
3683          POP IX
3684          LD 1Y, [WORK_BUFFER]
3685          LD HL, (1Y+F_GEN), A
3686          PUSH DE
3687          LD BC, XP_OS
3688          LD HL, [WORK_BUFFER]
3689          LD D, (1X+5)
3690          LD E, (1X+4)
3691          LD A, (1X+6)
3692          POP IX
3693          LD 1Y, [WORK_BUFFER]
3694          LD HL, (1Y+F_GEN), A
3695          PUSH DE
3696          LD BC, XP_OS
3697          LD HL, [WORK_BUFFER]
3698          LD D, (1X+5)
3699          LD E, (1X+4)
3700          LD A, (1X+6)
3701          POP IX
3702          LD 1Y, [WORK_BUFFER]
3703          LD HL, (1Y+F_GEN), A
3704          PUSH DE
3705          LD BC, XP_OS
3706          LD HL, [WORK_BUFFER]
3707          LD D, (1X+5)
3708          LD E, (1X+4)
3709          LD A, (1X+6)
3710          POP IX
3711          LD 1Y, [WORK_BUFFER]
3712          LD HL, (1Y+F_GEN), A
3713          PUSH DE
3714          LD BC, XP_OS
3715          LD HL, [WORK_BUFFER]
3716          LD D, (1X+5)
3717          LD E, (1X+4)
3718          LD A, (1X+6)
3719          POP IX
3720          LD 1Y, [WORK_BUFFER]
3721          LD HL, (1Y+F_GEN), A
3722          PUSH DE
3723          LD BC, XP_OS
3724          LD HL, [WORK_BUFFER]
3725          LD D, (1X+5)
3726          LD E, (1X+4)
3727          LD A, (1X+6)
3728          POP IX
3729          LD 1Y, [WORK_BUFFER]
3730          LD HL, (1Y+F_GEN), A
3731          PUSH DE
3732          LD BC, XP_OS
3733          LD HL, [WORK_BUFFER]
3734          LD D, (1X+5)
3735          LD E, (1X+4)
3736          LD A, (1X+6)
3737          POP IX
3738          LD 1Y, [WORK_BUFFER]
3739          LD HL, (1Y+F_GEN), A
3740          PUSH DE
3741          LD BC, XP_OS
3742          LD HL, [WORK_BUFFER]
3743          LD D, (1X+5)
3744          LD E, (1X+4)
3745          LD A, (1X+6)
3746          POP IX
3747          LD 1Y, [WORK_BUFFER]
3748          LD HL, (1Y+F_GEN), A
3749          PUSH DE
3750          LD BC, XP_OS
3751          LD HL, [WORK_BUFFER]
3752          LD D, (1X+5)
3753          LD E, (1X+4)
3754          LD A, (1X+6)
3755          POP IX
3756          LD 1Y, [WORK_BUFFER]
3757          LD HL, (1Y+F_GEN), A
3758          PUSH DE
3759          LD BC, XP_OS
3760          LD HL, [WORK_BUFFER]
3761          LD D, (1X+5)
3762          LD E, (1X+4)
3763          LD A, (1X+6)
3764          POP IX
3765          LD 1Y, [WORK_BUFFER]
3766          LD HL, (1Y+F_GEN), A
3767          PUSH DE
3768          LD BC, XP_OS
3769          LD HL, [WORK_BUFFER]
3770          LD D, (1X+5)
3771          LD E, (1X+4)
3772          LD A, (1X+6)
3773          POP IX
3774          LD 1Y, [WORK_BUFFER]
3775          LD HL, (1Y+F_GEN), A
3776          PUSH DE
3777          LD BC, XP_OS
3778          LD HL, [WORK_BUFFER]
3779          LD D, (1X+5)
3780          LD E, (1X+4)
3781          LD A, (1X+6)
3782          POP IX
3783          LD 1Y, [WORK_BUFFER]
3784          LD HL, (1Y+F_GEN), A
3785          PUSH DE
3786          LD BC, XP_OS
3787          LD HL, [WORK_BUFFER]
3788          LD D, (1X+5)
3789          LD E, (1X+4)
3790          LD A, (1X+6)
3791          POP IX
3792          LD 1Y, [WORK_BUFFER]
3793          LD HL, (1Y+F_GEN), A
3794          PUSH DE
3795          LD BC, XP_OS
3796          LD HL, [WORK_BUFFER]
3797          LD D, (1X+5)
3798          LD E, (1X+4)
3799          LD A, (1X+6)
3800          POP IX
3801          LD 1Y, [WORK_BUFFER]
3802          LD HL, (1Y+F_GEN), A
3803          PUSH DE
3804          LD BC, XP_OS
3805          LD HL, [WORK_BUFFER]
3806          LD D, (1X+5)
3807          LD E, (1X+4)
3808          LD A, (1X+6)
3809          POP IX
3810          LD 1Y, [WORK_BUFFER]
3811          LD HL, (1Y+F_GEN), A
3812          PUSH DE
3813          LD BC, XP_OS
3814          LD HL, [WORK_BUFFER]
3815          LD D, (1X+5)
3816          LD E, (1X+4)
3817          LD A, (1X+6)
3818          POP IX
3819          LD 1Y, [WORK_BUFFER]
3820          LD HL, (1Y+F_GEN), A
3821          PUSH DE
3822          LD BC, XP_OS
3823          LD HL, [WORK_BUFFER]
3824          LD D, (1X+5)
3825          LD E, (1X+4)
3826          LD A, (1X+6)
3827          POP IX
3828          LD 1Y, [WORK_BUFFER]
3829          LD HL, (1Y+F_GEN), A
3830          PUSH DE
3831          LD BC, XP_OS
3832          LD HL, [WORK_BUFFER]
3833          LD D, (1X+5)
3834          LD E, (1X+4)
3835          LD A, (1X+6)
3836          POP IX
3837          LD 1Y, [WORK_BUFFER]
3838          LD HL, (1Y+F_GEN), A
3839          PUSH DE
3840          LD BC, XP_OS
3841          LD HL, [WORK_BUFFER]
3842          LD D, (1X+5)
3843          LD E, (1X+4)
3844          LD A, (1X+6)
3845          POP IX
3846          LD 1Y, [WORK_BUFFER]
3847          LD HL, (1Y+F_GEN), A
3848          PUSH DE
3849          LD BC, XP_OS
3850          LD HL, [WORK_BUFFER]
3851          LD D, (1X+5)
3852          LD E, (1X+4)
3853          LD A, (1X+6)
3854          POP IX
3855          LD 1Y, [WORK_BUFFER]
3856          LD HL, (1Y+F_GEN), A
3857          PUSH DE
3858          LD BC, XP_OS
3859          LD HL, [WORK_BUFFER]
3860          LD D, (1X+5)
3861          LD E, (1X+4)
3862          LD A, (1X+6)
3863          POP IX
3864          LD 1Y, [WORK_BUFFER]
3865          LD HL, (1Y+F_GEN), A
3866          PUSH DE
3867          LD BC, XP_OS
3868          LD HL, [WORK_BUFFER]
3869          LD D, (1X+5)
3870          LD E, (1X+4)
3871          LD A, (1X+6)
3872          POP IX
3873          LD 1Y, [WORK_BUFFER]
3874          LD HL, (1Y+F_GEN), A
3875          PUSH DE
3876          LD BC, XP_OS
3877          LD HL, [WORK_BUFFER]
3878          LD D, (1X+5)
3879          LD E, (1X+4)
3880          LD A, (1X+6)
3881          POP IX
3882          LD 1Y, [WORK_BUFFER]
3883          LD HL, (1Y+F_GEN), A
3884          PUSH DE
3885          LD BC, XP_OS
3886          LD HL, [WORK_BUFFER]
3887          LD D, (1X+5)
3888          LD E, (1X+4)
3889          LD A, (1X+6)
3890          POP IX
3891          LD 1Y, [WORK_BUFFER]
3892          LD HL, (1Y+F_GEN), A
3893          PUSH DE
3894          LD BC, XP_OS
3895          LD HL, [WORK_BUFFER]
3896          LD D, (1X+5)
3897          LD E, (1X+4)
3898          LD A, (1X+6)
3899          POP IX
3900          LD 1Y, [WORK_BUFFER]
3901          LD HL, (1Y+F_GEN), A
3902          PUSH DE
3903          LD BC, XP_OS
3904          LD HL, [WORK_BUFFER]
3905          LD D, (1X+5)
3906          LD E, (1X+4)
3907          LD A, (1X+6)
3908          POP IX
3909          LD 1Y, [WORK_BUFFER]
3910          LD HL, (1Y+F_GEN), A
3911          PUSH DE
3912          LD BC, XP_OS
3913          LD HL, [WORK_BUFFER]
3914          LD D, (1X+5)
3915          LD E, (1X+4)
3916          LD A, (1X+6)
3917          POP IX
3918          LD 1Y, [WORK_BUFFER]
3919          LD HL, (1Y+F_GEN), A
3920          PUSH DE
3921          LD BC, XP_OS
3922          LD HL, [WORK_BUFFER]
3923          LD D, (1X+5)
3924          LD E, (1X+4)
3925          LD A, (1X+6)
3926          POP IX
3927          LD 1Y, [WORK_BUFFER]
3928          LD HL, (1Y+F_GEN), A
3929          PUSH DE
3930          LD BC, XP_OS
3931          LD HL, [WORK_BUFFER]
3932          LD D, (1X+5)
3933          LD E, (1X+4)
3934          LD A, (1X+6)
3935          POP IX
3936          LD 1Y, [WORK_BUFFER]
3937          LD HL, (1Y+F_GEN), A
3938          PUSH DE
3939          LD BC, XP_OS
3940          LD HL, [WORK_BUFFER]
3941          LD D, (1X+5)
3942          LD E, (1X+4)
3943          LD A, (1X+6)
3944          POP IX
3945          LD 1Y, [WORK_BUFFER]
3946          LD HL, (1Y+F_GEN), A
3947          PUSH DE
3948          LD BC, XP_OS
3949          LD HL, [WORK_BUFFER]
3950          LD D, (1X+5)
3951          LD E, (1X+4)
3952          LD A, (1X+6)
3953          POP IX
3954          LD 1Y, [WORK_BUFFER]
3955          LD HL, (1Y+F_GEN), A
3956          PUSH DE
3957          LD BC, XP_OS
3958          LD HL, [WORK_BUFFER]
3959          LD D, (1X+5)
3960          LD E, (1X+4)
3961          LD A, (1X+6)
3962          POP IX
3963          LD 1Y, [WORK_BUFFER]
3964          LD HL, (1Y+F_GEN), A
3965          PUSH DE
3966          LD BC, XP_OS
3967          LD HL, [WORK_BUFFER]
3968          LD D, (1X+5)
3969          LD E, (1X+4)
3970          LD A, (1X+6)
3971          POP IX
3972          LD 1Y, [WORK_BUFFER]
3973          LD HL, (1Y+F_GEN), A
3974          PUSH DE
3975          LD BC, XP_OS
3976          LD HL, [WORK_BUFFER]
3977          LD D, (1X+5)
3978          LD E, (1X+4)
3979          LD A, (1X+6)
3980          POP IX
3981          LD 1Y, [WORK_BUFFER]
3982          LD HL, (1Y+F_GEN), A
3983          PUSH DE
3984          LD BC, XP_OS
3985          LD HL, [WORK_BUFFER]
3986          LD D, (1X+5)
3987          LD E, (1X+4)
3988          LD A, (1X+6)
3989          POP IX
3990          LD 1Y, [WORK_BUFFER]
3991          LD HL, (1Y+F_GEN), A
3992          PUSH DE
3993          LD BC, XP_OS
3994          LD HL, [WORK_BUFFER]
3995          LD D, (1X+5)
3996          LD E, (1X+4)
3997          LD A, (1X+6)
3998          POP IX
3999          LD 1Y, [WORK_BUFFER]
4000          LD HL, (1Y+F_GEN), A
4001          PUSH DE
4002          LD BC, XP_OS
4003          LD HL, [WORK_BUFFER]
4004          LD D, (1X+5)
4005          LD E, (1X+4)
4006          LD A, (1X+6)
4007          POP IX
4008          LD 1Y, [WORK_BUFFER]
4009          LD HL, (1Y+F_GEN), A
4010          PUSH DE
4011          LD BC, XP_OS
4012          LD HL, [WORK_BUFFER]
4013          LD D, (1X+5)
4014          LD E, (1X+4)
4015          LD A, (1X+6)
4016          POP IX
4017          LD 1Y, [WORK_BUFFER]
4018          LD HL, (1Y+F_GEN), A
4019          PUSH DE
4020          LD BC, XP_OS
4021          LD HL, [WORK_BUFFER]
4022          LD D, (1X+5)
4023          LD E, (1X+4)
4024          LD A, (1X+6)
4025          POP IX
4026          LD 1Y, [WORK_BUFFER]
4027          LD HL, (1Y+F_GEN), A
4028          PUSH DE
4029          LD BC, XP_OS
4030          LD HL, [WORK_BUFFER]
4031          LD D, (1X+5)
4032          LD E, (1X+4)
4033          LD A, (1X+6)
4034          POP IX
4035          LD 1Y, [WORK_BUFFER]
4036          LD HL, (1Y+F_GEN), A
4037          PUSH DE
4038          LD BC, XP_OS
4039          LD HL, [WORK_BUFFER]
4040          LD D, (1X+5)
4041          LD E, (1X+4)
4042          LD A, (1X+6)
4043          POP IX
4044          LD 1Y, [WORK_BUFFER]
4045          LD HL, (1Y+F_GEN), A
4046          PUSH DE
4047          LD BC, XP_OS
4048          LD HL, [WORK_BUFFER]
4049          LD D, (1X+5)
4050          LD E, (1X+4)
4051          LD A, (1X+6)
4052          POP IX
4053          LD 1Y, [WORK_BUFFER]
4054          LD HL, (1Y+F_GEN), A
4055          PUSH DE
4056          LD BC, XP_OS
4057          LD HL, [WORK_BUFFER]
4058          LD D, (1X+5)
4059          LD E, (1X+4)
4060          LD A, (1X+6)
4061          POP IX
4062          LD 1Y, [WORK_BUFFER]
4063          LD HL, (1Y+F_GEN), A
4064          PUSH DE
4065          LD BC, XP_OS
4066          LD HL, [WORK_BUFFER]
4067          LD D, (1X+5)
4068          LD E, (1X+4)
4069          LD A, (1X+6)
4070          POP IX
4071          LD 1Y, [WORK_BUFFER]
4072          LD HL, (1Y+F_GEN), A
4073          PUSH DE
4074          LD BC, XP_OS
4075          LD HL, [WORK_BUFFER]
4076          LD D, (1X+5)
4077          LD E, (1X+4)
4078          LD A, (1X+6)
4079          POP IX
4080          LD 1Y, [WORK_BUFFER]
4081          LD HL, (1Y+F_GEN), A
4082          PUSH DE
4083          LD BC, XP_OS
4084          LD HL, [WORK_BUFFER]
4085          LD D, (1X+5)
4086          LD E, (1X+4)
4087          LD A, (1X+6)
4088          POP IX
4089          LD 1Y, [WORK_BUFFER]
4090          LD HL, (1Y+F_GEN), A
4091          PUSH DE
4092          LD BC, XP_OS
4093          LD HL, [WORK_BUFFER]
4094          LD D, (1X+5)
4095          LD E, (1X+4)
4096          LD A, (1X+6)
4097          POP IX
4098          LD 1Y, [WORK_BUFFER]
4099          LD HL, (1Y+F_GEN), A
4100          PUSH DE
4101          LD BC,
```

LOCATION	OBJECT CODE	LINE	SOURCE LINE
0AF9 09		3273	ADD HL,BC
0AFA 010008		3274	LD BC,11
		3275 ;	IF [D,LT,70H]
0AFD 7A		3276	LD A,D
0AFE FE70		3277	CP 70H
0800 3005		3278	JR NC,ELSE2
0802 CD1FE2		3279	CALL READ_VRAM
0805 1803		3280	JR END2
		3281 ;	ELSE
0807 EB		3282	EX DE,HL
0808 ED80		3283	LDIR
080A		3284	END2 ;ENDIF
		3285	
080A 2A8006		3286 ;	AT THIS POINT, IX = GRAPHICS, [SP] = OLD SCREEN
080B 110013		3287 ;	BACKGROUND PATTERN POSITION AND NAMES STARTING AT YP_BK
0810 19		3288 ;	OLD SCREEN PATTERN POSITION AND NAMES STARTING AT YP_OS
0811 D9		3289 ;	FIND ALL NAMES IN BACKGROUND WHICH BELONG TO THIS OBJECT'S PATTERN GENERATORS
0812 ED5B006		3290 ;	AND REPLACE WITH NAME FROM OLD_SCREEN WHICH CORRESPONDS TO THAT PATTERN POSITION
0816 210008		3291 PH3	LD HL,[WORK_BUFFER]
0819 19		3292	LD DE,YP_BK+1
081A EB		3293	ADD HL,DE
081B D9		3294	EXX
081C FD2A006		3295	LD DE,[WORK_BUFFER]
0820 FD4E05		3296	LD HL,YP_OS+1
		3297	ADD HL,DE
		3298	EX DE,HL
		3299	EXX
		3300	LD IY,[WORK_BUFFER]
		3301	LD C,[IY+GEN]
		3302 ;	DO B,9
		3303	LD B,9
0823 0609		3304	LD A,[HL]
0825 7E		3305	SUB C
0826 91		3306 ;	IF [A,LT,18]
0827 FE12		3307	CP 18
0829 300E		3308	JR NC,END3
		3309 ;	IF [A,GE,9]
082B FE09		3310	CP 9
082D 3802		3311	JR C,END4
082F 0609		3312	SUB 9
0831		3313	ENDIF
0831 D9		3314	EXX
0832 6F		3315	LD L,A
0833 2600		3316	LD H,0
0835 19		3317	ADD HL,DE
0836 7E		3318	LD A,[HL]
0837 D9		3319	EXX
0838 77		3320	LD [HL],A
0839		3321	ENDIF
0839 23		3322	INC HL
		3323 ;	ENDDO
083A 10E9		3324	DJNZ DLP1
		3325	
083C D1		3326 ;	NOW NEW VERSION OF BACKGROUND NAMES WILL NOT CONTAIN ANY NAMES OF THIS OBJECT
083D 2A8006		3327 ;	REPLACE PREVIOUS VERSION OF OLD_SCREEN WITH THIS NEW BACKGROUND
		3328 PH4	POP DE
		3329	LD HL,[WORK_BUFFER]

;GET 9 NAMES FROM VRAM
;THEN OLD_SCREEN IS IN VRAM

;OLD_SCREEN IN CPU RAM

;POINT TO FIRST OF BACKGROUND NAMES

;DE' := BUFFER BASE

;POINTS TO FIRST OF OLD_SCREEN NAMES

;C := FIRST_GEN_NAME

;GET A NAME

;SUBTRACT FIRST_GEN_NAME

;THEN NAME FALLS IN RANGE OF NAMES FOR OBJECT

;THEN SUB 9 TO FIND CORRECT

; POSITION IN OLD_SCREEN

;FORM A POINTER INTO OLD_SCREEN NAMES

;GET OLD_SCREEN NAME

;REPLACE BACKGROUND NAME WITH OLD_SCREEN NAME

;POINT TO NEXT NAME IN BACKGROUND

;NOW NEW VERSION OF BACKGROUND NAMES WILL NOT CONTAIN ANY NAMES OF THIS OBJECT

;REPLACE PREVIOUS VERSION OF OLD_SCREEN WITH THIS NEW BACKGROUND

;DE := OLD_SCREEN ADDRESS

;HL := BUFFER BASE