

```
video:3C00          ;
video:3C00          ;
Eiffiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii»
video:3C00          ; °      This file is generated by The Interactive Disassembler (IDA)
°
video:3C00          ; °      Licensed to: Unknown User ;-)
°
video:3C00          ; °      Copyright (c) 1999 by DataRescue sa/nv, <ida@datarescue.com>
°
video:3C00          ;
Eiffiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii»
video:3C00          ;
video:3C00          ; Processor:      z80
video:3C00          ; Target assembler: ASxxxx by Alan R. Baldwin v1.5
video:3C00          .area idaseg (ABS)
video:3C00          .hd64 ; this is needed only for HD64180
video:3C00          ;
ffffiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
video:3C00          ; Segment type: Regular
video:3C00          ; segment 'video'
video:3C00          .org 0x3C00
video:3C00 ?? ?? ?? ??+video_ram:.ds 0x400          ; DATA XREF:
update_bullet+84|o
video:3C00 ?? ?? ?? ??+          ; update_bullet+91|o ...
video:3C00 ?? ?? ?? ??+; end of 'video'
video:3C00 ?? ?? ?? ??+
ram:4300            ;
ffffiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
ram:4300
ram:4300            ; Segment type: Regular
ram:4300            ; segment 'ram'
ram:4300            .org 0x4300
ram:4300 ??         fire_throttle:.ds 1          ; DATA XREF: handle_fire+5|r
ram:4300            ; handle_fire+C|w ...
ram:4301 ??         invaders_left:.ds 1          ; DATA XREF: update_bullet+62|r
ram:4301            ; update_bullet+69|w ...
ram:4302 ?? ??     row_1_invader_addr:.ds 2      ; DATA XREF:
animate_invaders+2|r
ram:4302            ;
find_end_of_lowest_invader_row+15|r ...
ram:4304 ?? ??     row_2_invader_addr:.ds 2      ; DATA XREF:
animate_invaders+13|r
ram:4304            ;
find_end_of_lowest_invader_row+E|r ...
ram:4306 ?? ??     row_3_invader_addr:.ds 2      ; DATA XREF:
animate_invaders+19|r
ram:4306            ;
find_end_of_lowest_invader_row+7|r ...
ram:4308 ?? ??     row_4_invader_addr:.ds 2      ; DATA XREF:
animate_invaders+1F|r
ram:4308            ;
find_end_of_lowest_invader_row|r ...
ram:430A*??        invader_dir:.ds 1          ; DATA XREF: code:49DC|w
ram:430A*          ; animate_and_move_invaders+3|r
...
ram:430B*??        ufo_TTL:.ds 1          ; DATA XREF:
check_and_start_ufo+16|w
ram:430B*          ; update_ufo+35|r ...
ram:430C*??        ufo_timer:.ds 1          ; DATA XREF: update_bullet+C9|w
ram:430C*          ; check_and_start_ufo|r ...
ram:430D ??        ufo_dir:.ds 1          ; DATA XREF: handle_fire+15|r
ram:430D            ; handle_fire+1A|w ...
ram:430D            ; something with UFO
(direction?)
ram:430E*??        wave_no:.ds 1          ; DATA XREF: code:49CF|w
ram:430E*          ; code:49FC|r ...
ram:430F ??        no_lives:.ds 1          ; DATA XREF:
update_score_and_chk_bonus_life+1D|r
ram:430F            ;
update_score_and_chk_bonus_life+21|w ...
ram:430F            ; end of 'ram'
ram:430F
```

```
code:4310* ;File Name : tandy.bin
code:4310* ; Format : Binary File
code:4310* ; Base Address: 0000h Range: 4310h - 5000h Loaded length: 0CF0h
code:4310* ;
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
code:4310*
code:4310* ; Segment type: Pure code
code:4310* ; segment 'code'
code:4310* .org 0x4310
code:4310*00 ufo_active:.db 0 ; DATA XREF: handle_fire+F|r
code:4310* ; update_bullet+C6|w ...
code:4311*00 bullet_active:.db 0 ; DATA XREF: handle_fire|r
code:4311* ; handle_fire+1F|w ...
code:4312*C9 unused_4312:.db 0xC9 ; DATA XREF: code:49C9|w
code:4312* ; code:4AC1|w ...
code:4313*B0 45 base_centre:.dw 0x45B0 ; DATA XREF: handle_fire+23|r
code:4313* ; check_and_handle_move+17|r
...
code:4315*01 unused_4315:.db 1 ; DATA XREF: code:49C6|w
code:4316 40 3C init_row_1_invader_addr:.dw video_ram+0x40 ; DATA XREF: code:4A1E|o
code:4318 C0 3C init_row_2_invader_addr:.dw video_ram+0xC0
code:431A 40 3D init_row_3_invader_addr:.dw video_ram+0x140
code:431C C0 3D init_row_4_invader_addr:.dw video_ram+0x1C0
code:431E*53 3E bullet_addr:.dw video_ram+0x253 ; DATA XREF: handle_fire+2A|w
code:431E* ; update_bullet+1|r ...
code:4320*60 game_timer:.db 0x60 ; DATA XREF: code:4F02|w
code:4320* ; code:4F26|r ...
code:4321*2B invader_timer:.db 0x2B ; DATA XREF: code:4A98|w
code:4321* ; code:4F6C|r ...
code:4322*00 keybd_state:.db 0 ; DATA XREF: code:49D8|w
code:4322* ; code:4F17|r ...
code:4323 A0 B6 BF B9+invader_30pt:.db 0xA0, 0xB6, 0xBF, 0xB9, 0x90, 0x1A, 1, 0x20, 0x86,
0x20, 0x89 ; DATA XREF: code:4570|o
code:4323 90 1A 01 20+ ; code:4A72|o
code:4323 86 20 89 20+
code:4323 00 .db 0x20, 0
code:4330 9C B7 BF BB+invader_20pt:.db 0x9C, 0xB7, 0xBF, 0xBB, 0xAC, 0x1A, 1, 0x8C, 0x83,
0x20, 0x83 ; DATA XREF: code:457C|o
code:4330 AC 1A 01 8C+ ; code:4A7B|o
code:4330 83 20 83 8C+
code:4330 00 .db 0x8C, 0
code:433D BE BB BF B7+invader_10pt:.db 0xBE, 0xBB, 0xBF, 0xB7, 0xBD, 0x1A, 1, 0x8C, 0x83,
0x20, 0x83 ; DATA XREF: code:4588|o
code:433D BD 1A 01 8C+ ; code:4A84|o
code:433D 83 20 83 8C+
code:433D 00 .db 0x8C, 0
code:434A 8C B7 B7 B7+ufo: .db 0x8C, 0xB7, 0xB7, 0xB7, 0xB7, 0x8C, 0
code:434A B7 8C 00 ; DATA XREF: code:4594|o
code:434A ; check_and_start_ufo+2A|o
code:4351 B8 BF BF BF+shield: .db 0xB8, 0xBF, 0xBF, 0xBF, 0xBF, 0xBF, 0xB4, 0x1A, 8, 8, 1,
0x8F ; DATA XREF: code:4A4B|o
code:4351 BF BF B4 1A+ ; DATA XREF: code:4A4B|o
code:4351 08 08 01 8F+ .db 0x8F, 0x83, 0x83, 0x83, 0x8F, 0x8F, 0
code:4364 B8 BC BF BC+player: .db 0xB8, 0xBC, 0xBF, 0xBC, 0xB4, 0 ; DATA XREF: code:4AA1|o
code:436A 82 84 20 88+explosion:.db 0x82, 0x84, 0x20, 0x88, 0x81, 0x1A, 1, 0x88, 0x81, 0x20,
0x82 ; DATA XREF: update_bullet+53|o
code:436A 81 1A 01 88+ ; DATA XREF: update_bullet+53|o
code:436A 81 20 82 84+ .db 0x84, 0
code:4377 00 .db 0 ;
code:4378 00 .db 0 ;
code:4379 1B 01 20 20+blank_space:.db 0x1B, 1, 0x20, 0x20, 0x20, 0x20, 0x20, 0x1A, 1, 0x20,
0x20 ; DATA XREF: update_bullet+5C|o
code:4379 20 20 20 1A+ ; DATA XREF: update_bullet+5C|o
code:4379 01 20 20 20+ .db 0x20, 0x20, 0x20, 0
code:4388 50 4C 41 59 aPlay: .ascii 'PLAY' ; DATA XREF: code:4564|o
code:438C 09 D6 3C .db 9, 0xD6, 0x3C
code:438F 54 41 4E 44+aTandyInvaders:.ascii 'TANDY INVADERS'
code:43A3 09 54 3D .db 9, 0x54, 0x3D
code:43A6 2A 20 53 43+aScoreAdvanceTa:.ascii '* SCORE ADVANCE TABLE *'
code:43A6 4F 52 45 20+ .db 0
code:43BE 2A 20 54 41+aTandyElectroni:.ascii '* TANDY ELECTRONICS *' ; DATA XREF: code:4558|o
code:43BE 4E 44 59 20+ ; code:45BB|o
code:43BE 45 4C 45 43+ .db 0
code:43D4 3C 2D 2D 2D+a30Points:.ascii '<---- 30 POINTS' ; DATA XREF: code:45A0|o
code:43E6 09 1E 3E .db 9, 0x1E, 0x3E
code:43E9 3C 2D 2D 2D+a20Points:.ascii '<---- 20 POINTS'
code:43FB 09 9E 3E .db 9, 0x9E, 0x3E
```

[illegible]

```

code:4506 B7                or      a                    ; done?
code:4507 C8                ret     Z                    ; yes, exit
code:4508 D5                push    de
code:4509 C5                push    bc
code:450A CD 33 00          call    0x33                    ; display character
code:450D 01 00 23          ld       bc, #8960              ; ~130ms
code:4510 CD 60 00          call    0x60                    ; delay
code:4513 C1                pop     bc
code:4514 D1                pop     de
code:4515 23                inc     hl                    ; next character
code:4516 18 ED             jr       display_message_slowly ; loop through message
code:4516                ; End of function display_message_slowly
code:4516
code:4518
code:4518                ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000
code:4518
code:4518
code:4518                wipe_screen_left_to_right_slow:        ; CODE XREF: code:454F|p
code:4518 D9                                ; code:45AF|p ...
code:4518                exx
code:4519 21 FF 3B          ld       hl, #0x3BFF            ; start of video (-1)
code:451C 06 40            ld       b, #0x40 ; '@'           ; characters/line
code:451E                loc_0_451E:                            ; CODE XREF:
wipe_screen_left_to_right_slow+19|j
code:451E C5                push    bc
code:451F 23                inc     hl                    ; next column
code:4520 E5                push    hl
code:4521 06 0F            ld       b, #0xF               ; 15 lines
code:4523 3E 20            ld       a, #0x20 ; ' '         ; space
code:4525 11 40 00          ld       de, #0x40 ; '@'       ; next line address
code:4528                loc_0_4528:                            ; CODE XREF:
wipe_screen_left_to_right_slow+12|j
code:4528 77                ld       (hl), a              ; display character
code:4529 19                add     hl, de                 ; next line
code:452A 10 FC            djnz     loc_0_4528          ; loop all lines
code:452C CD 60 4E          call    delay_1_5ms
code:452F E1                pop     hl
code:4530 C1                pop     bc
code:4531 10 EB            djnz     loc_0_451E          ; loop all columns
code:4533 D9                exx
code:4534 C9                ret
code:4534                ; End of function wipe_screen_left_to_right_slow
code:4534
code:4535                ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4535
code:4535                START:
code:4535 F3                di
code:4536 3E 0F            ld       a, #0xF
code:4538 CD 33 00          call    0x33                    ; display character
code:453B 31 8A 42          ld       sp, #0x428A
code:453E 21 C0 3F          ld       hl, #video_ram+0x3C0   ; cursor position
code:4541 22 20 40          ld       (0x4020), hl
code:4544 21 A5 44          ld       hl, #aScore00000High ; " SCORE 00000
"...
code:4547 CD E5 44          call    display_message
code:454A 3E 20            ld       a, #0x20 ; ' '
code:454C 32 FF 3F          ld       (video_ram+0x3FF), a
code:454F                attract_loop:                            ; CODE XREF: code:45D6|j
code:454F CD 18 45                                ; code:4960|j
code:454F                call    wipe_screen_left_to_right_slow
code:4552 21 94 3F          ld       hl, #video_ram+0x394   ; cursor position
code:4555 22 20 40          ld       (0x4020), hl
code:4558 21 BE 43          ld       hl, #aTandyElectroni ; "* TANDY ELECTRONICS *"
code:455B CD E5 44          call    display_message
code:455E 21 9E 3C          ld       hl, #video_ram+0x9E    ; cursor position
code:4561 22 20 40          ld       (0x4020), hl
code:4564 21 88 43          ld       hl, #aPlay            ; "PLAY"
code:4567 CD 66 4E          call    print_slow_and_check_for_R_key
code:456A 21 92 3D          ld       hl, #video_ram+0x192   ; cursor position
code:456D 22 20 40          ld       (0x4020), hl
code:4570 21 23 43          ld       hl, #invader_30pt
code:4573 CD E5 44          call    display_message

```

[illegible]

```

code:45EE          loc_0_45EE:                                ; CODE XREF:
compare_video_addresses+4↑j
code:45EE 7D          ld      a, 1
code:45EF BB          cp      e                                ; same address?
code:45F0 28 06       jr      Z, loc_0_45F8                    ; yes, continue
code:45F2 38 F6       jr      C, no_hit                        ; exit if bullet above
code:45F4          loc_0_45F4:                                ; CODE XREF:
compare_video_addresses+6↑j
code:45F4 3E FF       ld      a, #0xFF                          ; flag bullet right/below
code:45F6 18 F3       jr      loc_0_45EB
code:45F8          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:45F8          loc_0_45F8:                                ; CODE XREF:
compare_video_addresses+E↑j
code:45F8 3E 20       ld      a, #0x20 ; ' '                      ; flag bullet match
code:45FA 18 EF       jr      loc_0_45EB
code:45FA          ; End of function compare_video_addresses
code:45FC          ; 0000000000000000 S U B R O U T I N E
000000000000000000000000000000000000000000000000000000000000
code:45FC
code:45FC          animate_invaders:                          ; CODE XREF:
animate_and_move_invaders+2B↑p
code:45FC E5          ; code:4B91↑p
code:45FC          push     hl
code:45FD C5          push     bc
code:45FE 2A 02 43    ld      hl, (row_1_invader_addr)
code:4601 06 3F       ld      b, #0x3F ; '?'                      ; characters/line-1
code:4603          animate_top_row:                          ; CODE XREF:
animate_invaders+10↑j
code:4603 7E          ld      a, (hl)                      ; get character from video
code:4604 CB 7F       bit      7, a                          ; graphic character?
code:4606 28 03       jr      Z, loc_0_460B                    ; no, skip
code:4608 EE 0F       xor      #0xF                          ; invert top 4 pixels
code:460A 77          ld      (hl), a                          ; display
code:460B          loc_0_460B:                                ; CODE XREF:
animate_invaders+A↑j
code:460B 23          inc      hl                          ; next video address
code:460C 10 F5       djnz    animate_top_row                    ; loop thru line
code:460E D5          push     de
code:460F 2A 04 43    ld      hl, (row_2_invader_addr)
code:4612 CD 25 46    call    animate_invader_row
code:4615 2A 06 43    ld      hl, (row_3_invader_addr)
code:4618 CD 25 46    call    animate_invader_row
code:461B 2A 08 43    ld      hl, (row_4_invader_addr)
code:461E CD 25 46    call    animate_invader_row
code:4621 D1          pop      de
code:4622 C1          pop      bc
code:4623 E1          pop      hl
code:4624 C9          ret
code:4624          ; End of function animate_invaders
code:4625          ; 0000000000000000 S U B R O U T I N E
000000000000000000000000000000000000000000000000000000000000
code:4625          animate_invader_row:                      ; CODE XREF:
animate_invaders+16↑p
code:4625 E5          ; animate_invaders+1C↑p ...
code:4625          push     hl
code:4626 06 40       ld      b, #0x40 ; '@'                      ; characters/line
code:4628          loc_0_4628:                                ; CODE XREF:
animate_invader_row+15↑j
code:4628 7E          ld      a, (hl)                      ; get character from video
code:4629 FE 8C       cp      #0x8C ; 'E'
code:462B 28 09       jr      Z, loc_0_4636
code:462D FE 83       cp      #0x83 ; 'f'
code:462F 20 08       jr      NZ, loc_0_4639
code:4631 3E 8C       ld      a, #0x8C ; 'E'                      ; not part of animation, skip
; 0x83->0x8C

```

```

code:4633 77          ld      (hl), a          ; display
code:4634 18 03          jr      loc_0_4639
code:4636          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4636
code:4636          loc_0_4636:                ; CODE XREF:
animate_invader_row+6↑j
code:4636 3E 83          ld      a, #0x83 ; 'f'          ; 0x8C->0x83
code:4638 77          ld      (hl), a          ; display
code:4639
code:4639          loc_0_4639:                ; CODE XREF:
animate_invader_row+A↑j
code:4639 23          ; animate_invader_row+F↑j
code:4639          inc      hl                ; next video address
code:463A 10 EC          djnz   loc_0_4628          ; loop thru line
code:463C E1          pop      hl
code:463D C9          ret
code:463D          ; End of function animate_invader_row
code:463D
code:463E          ; 0000000000000000 S U B R O U T I N E
000000000000000000000000000000000000000000000000000000000000
code:463E
code:463E
code:463E          move_video_line_right_HL:        ; CODE XREF:
check_and_handle_move+14↑p
code:463E E5          ; update_ufo+2A↑p ...
code:463E          push     hl
code:463F 11 3E 00          ld      de, #0x3E ; '>'          ; start at right end
code:4642 19          add      hl, de
code:4643 E5          push     hl
code:4644 D1          pop      de
code:4645 13          inc      de          ; DE = end of row
code:4646 06 3F          ld      b, #63          ; number of characters/line-1
code:4648 1A          ld      a, (de)          ; get character from end of row
code:4649 CB 7F          bit      7, a          ; graphic?
code:464B 28 03          jr      Z, loc_0_4650          ; no, skip
code:464D 3E 20          ld      a, #0x20 ; ' '          ; space
code:464F 12          ld      (de), a          ; display space
code:4650
code:4650          loc_0_4650:                ; CODE XREF:
move_video_line_right_HL+D↑j
code:4650 CB 7E          ; move_video_line_right_HL+1C↑j
code:4650          bit      7, (hl)          ; graphic character left byte?
code:4652 28 04          jr      Z, loc_0_4658          ; no, skip
code:4654 7E          ld      a, (hl)          ; get character left byte
code:4655 12          ld      (de), a          ; display to the right
code:4656 36 20          ld      (hl), #0x20 ; ' '          ; display space at left-hand
byte
code:4658
code:4658          loc_0_4658:                ; CODE XREF:
move_video_line_right_HL+14↑j
code:4658 2B          dec      hl
code:4659 1B          dec      de
code:465A 10 F4          djnz   loc_0_4650          ; loop through line
code:465C E1          pop      hl
code:465D 36 20          ld      (hl), #0x20 ; ' '          ; display space
code:465F C9          ret
code:465F          ; End of function move_video_line_right_HL
code:465F
code:4660
code:4660          ; 0000000000000000 S U B R O U T I N E
000000000000000000000000000000000000000000000000000000000000
code:4660
code:4660
code:4660          move_video_line_left_HL:        ; CODE XREF:
check_and_handle_move+2B↑p
code:4660 E5          ; update_ufo+32↑p ...
code:4660          push     hl
code:4661 E5          push     hl
code:4662 D1          pop      de
code:4663 23          inc      hl
code:4664 06 3F          ld      b, #63          ; characters/line-1
code:4666 1A          ld      a, (de)          ; get character LH byte
code:4667 CB 7F          bit      7, a          ; graphic?
code:4669 28 03          jr      Z, loc_0_466E          ; no, skip
code:466B 3E 20          ld      a, #0x20 ; ' '          ; space

```

```

code:466D 12          ld      (de), a           ; display space LH byte
code:466E             loc_0_466E:               ; CODE XREF:
code:466E             move_video_line_left_HL+9↑j                ;
code:466E CB 7E       ; move_video_line_left_HL+18↑j
code:466E             bit      7, (hl)         ; graphic RH byte?
code:4670 28 04       jr      Z, loc_0_4676    ; no, skip
code:4672 7E         ld      a, (hl)         ; get character from video (RH
byte)
code:4673 12          ld      (de), a           ; display LH byte
code:4674 36 20       ld      (hl), #0x20 ; ' '   ; space RH byte
code:4676             loc_0_4676:               ; CODE XREF:
code:4676             move_video_line_left_HL+10↑j              ;
code:4676 23          inc     hl
code:4677 13          inc     de
code:4678 10 F4       djnz   loc_0_466E        ; loop thru row
code:467A E1          pop     hl
code:467B 11 3F 00    ld      de, #0x3F ; '?'
code:467E 19          add     hl, de
code:467F 36 20       ld      (hl), #0x20 ; ' '   ; display space on end of row
code:4681 C9          ret
code:4681            ; End of function move_video_line_left_HL
code:4681
code:4682
code:4682            ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000
code:4682
code:4682
code:4682            add_10_to_score:           ; CODE XREF:
code:4682            update_score_and_chk_bonus_life↑p
code:4682 E5          push    hl
code:4683 C5          push    bc
code:4684 21 CC 3F     ld      hl, #video_ram+0x3CC ; tens digit
code:4687 CD A4 46     call    add_1_to_score_digit
code:468A 20 15       jr      NZ, loc_0_46A1    ; skip if no carry
code:468C 2B          dec     hl                ; hundreds digit
code:468D CD A4 46     call    add_1_to_score_digit
code:4690 20 0F       jr      NZ, loc_0_46A1    ; skip if no carry
code:4692 2B          dec     hl                ; thousands digit
code:4693 CD A4 46     call    add_1_to_score_digit
code:4696 20 09       jr      NZ, loc_0_46A1    ; skip if no carry
code:4698 2B          dec     hl                ; tens of thousands digit
code:4699 CD A4 46     call    add_1_to_score_digit
code:469C 20 03       jr      NZ, loc_0_46A1    ; skip if no carry
code:469E CD B0 46     call    zero_score
code:46A1            loc_0_46A1:                 ; CODE XREF:
code:46A1            add_10_to_score+8↑j
code:46A1 C1          ; add_10_to_score+E↑j ...
code:46A1             pop     bc
code:46A2 E1          pop     hl
code:46A3 C9          ret
code:46A3            ; End of function add_10_to_score
code:46A3
code:46A4
code:46A4            ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000
code:46A4
code:46A4
code:46A4            add_1_to_score_digit:       ; CODE XREF:
code:46A4            add_10_to_score+5↑p
code:46A4 7E         ; add_10_to_score+B↑p ...
code:46A4             ld      a, (hl)         ; get score digit
code:46A5 FE 39       cp      #0x39 ; '9'        ; 9?
code:46A7 28 03       jr      Z, flag_carry     ; yes, skip
code:46A9 3C          inc     a                ; add 1
code:46AA 77          ld      (hl), a          ; store
code:46AB C9          ret
code:46AC            ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:46AC            flag_carry:                 ; CODE XREF:
code:46AC            add_1_to_score_digit+3↑j
code:46AC 36 30       ld      (hl), #0x30 ; '0'    ; set to 0
code:46AE AF          xor     a                ; flag carry
code:46AF C9          ret

```



```

code:46AF          ; End of function add_1_to_score_digit
code:46AF
code:46B0
code:46B0          ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000
code:46B0
code:46B0
code:46B0          zero_score:                                ; CODE XREF:
add_10_to_score+1C|p
code:46B0 21 C9 3F          ld      hl, #video_ram+0x3C9      ; code:49EF|p
code:46B0          ld      hl, #5                             ; score
code:46B3 06 05          ld      b, #5                         ; 5 digits to zap
code:46B5          loc_0_46B5:                                ; CODE XREF: zero_score+8|j
code:46B5 36 30          ld      (hl), #0x30 ; '0'             ; set digit to 0
code:46B7 23          inc      hl                             ; next digit
code:46B8 10 FB          djnz     loc_0_46B5                  ; loop thru all score digits
code:46BA C9          ret
code:46BA          ; End of function zero_score
code:46BA
code:46BB          ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000
code:46BB
code:46BB          update_score_and_chk_bonus_life:          ; CODE XREF:
update_score_and_chk_bonus_life+27|j
code:46BB CD 82 46          ; update_bullet+50|p ...
code:46BB          call     add_10_to_score
code:46BE 21 CC 3F          ld      hl, #video_ram+0x3CC      ; tens digit of score
code:46C1 7E          ld      a, (hl)                        ; get digit
code:46C2 FE 30          cp      #0x30 ; '0'                     ; 0?
code:46C4 20 1C          jr      NZ, loc_0_46E2                ; no, exit
code:46C6 2B          dec      hl                             ; hundreds digit
code:46C7 7E          ld      a, (hl)                        ; get digit
code:46C8 FE 35          cp      #0x35 ; '5'                     ; 5?
code:46CA 20 16          jr      NZ, loc_0_46E2                ; no, skip
code:46CC 2B          dec      hl                             ; thousands digit
code:46CD 7E          ld      a, (hl)                        ; get digit
code:46CE FE 31          cp      #0x31 ; '1'                     ; 1?
code:46D0 20 10          jr      NZ, loc_0_46E2                ; no, skip
code:46D2 2B          dec      hl                             ; tens of thousands digit
code:46D3 7E          ld      a, (hl)                        ; get digit
code:46D4 FE 30          cp      #0x30 ; '0'                     ; 0?
code:46D6 20 0A          jr      NZ, loc_0_46E2                ; no, skip
code:46D8 3A 0F 43          ld      a, (no_lives)
code:46DB 3C          inc      a                             ; bonus life
code:46DC 32 0F 43          ld      (no_lives), a
code:46DF CD 90 4E          call    display_lives_left
code:46E2
code:46E2          loc_0_46E2:                                ; CODE XREF:
update_score_and_chk_bonus_life+9|j
code:46E2 10 D7          ;
update_score_and_chk_bonus_life+F|j ...
code:46E2          djnz     update_score_and_chk_bonus_life
code:46E4 C9          ret
code:46E4          ; End of function update_score_and_chk_bonus_life
code:46E4
code:46E5          ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000
code:46E5
code:46E5          handle_fire:                                ; CODE XREF: code:4F23|p
code:46E5 3A 11 43          ld      a, (bullet_active)
code:46E8 B7          or      a                             ; already fired?
code:46E9 C0          ret      NZ                             ; yes, return
code:46EA 3A 00 43          ld      a, (fire_throttle)
code:46ED B7          or      a                             ; waiting for throttle?
code:46EE C0          ret      NZ                             ; yes, return
code:46EF 3E 14          ld      a, #20                             ; init throttle value
code:46F1 32 00 43          ld      (fire_throttle), a
code:46F4 3A 10 43          ld      a, (ufo_active)
code:46F7 B7          or      a                             ; on-screen?
code:46F8 20 08          jr      NZ, loc_0_4702                ; yes, skip
code:46FA 3A 0D 43          ld      a, (ufo_dir)
code:46FD EE 01          xor      #1                             ; toggle direction

```

[illegible]

```

code:476B          loc_0_476B:                ; CODE XREF: update_bullet+4C↑j
code:476B 05          dec          b
code:476C
code:476C          loc_0_476C:                ; CODE XREF: update_bullet+4A↑j
code:476C CD BB 46    call      update_score_and_chk_bonus_life
code:476F 21 6A 43    ld        hl, #explosion
code:4772 CD E5 44    call      display_message
code:4775 CD 5A 4E    call      delay_15ms
code:4778 21 79 43    ld        hl, #blank_space
code:477B CD E5 44    call      display_message
code:477E 3A 01 43    ld        a, (invaders_left)
code:4781 3D          dec          a                ; end of wave?
code:4782 CA 74 48    jp        Z, end_of_wave        ; yes, go
code:4785 32 01 43    ld        (invaders_left), a
code:4788 11 00 00    ld        de, #0
code:478B CD 85 49    call      update_invader_row_addresses
code:478E 18 06          jr        clear_bullet_active
code:4790          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4790
code:4790          delete_bullet:                ; CODE XREF: update_bullet+11↑j
code:4790 11 40 00    ld        de, #0x40 ; '@'        ; characters/line
code:4793 19          add          hl, de                ; next row
code:4794 36 20    ld        (hl), #0x20 ; ' '        ; display space
code:4796
code:4796          clear_bullet_active:                ; CODE XREF: update_bullet+72↑j
code:4796 AF          ; update_bullet+82↑j ...
code:4796          xor          a
code:4797 32 11 43    ld        (bullet_active), a        ; clear fired flag
code:479A D9          exx
code:479B C9          ret
code:479C          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:479C
code:479C          check_and_handle_ufo_hit:                ; CODE XREF: update_bullet+2A↑j
code:479C CB 7E          bit        7, (hl)                ; graphic character?
code:479E 28 F6          jr        Z, clear_bullet_active ; no, hit a bomb, exit
code:47A0 21 00 3C    ld        hl, #video_ram
code:47A3 06 40    ld        b, #64                ; characters/line
code:47A5
code:47A5          loc_0_47A5:                ; CODE XREF: update_bullet+8E↑j
code:47A5 CB 7E          bit        7, (hl)                ; graphic character?
code:47A7 20 03          jr        NZ, loc_0_47AC          ; yes, skip
code:47A9 23          inc          hl                ; next video address
code:47AA 10 F9          djnz      loc_0_47A5              ; loop thru line
code:47AC
code:47AC          loc_0_47AC:                ; CODE XREF: update_bullet+8B↑j
code:47AC E5          push       hl
code:47AD 21 00 3C    ld        hl, #video_ram
code:47B0 CD 7E 4C    call      clear_video_line_HL        ; wipe UFO
code:47B3 21 06 00    ld        hl, #6                ; RAND(1-6)
code:47B6 CD 1D 4D    call      rand
code:47B9 45          ld        b, 1                ; get result
code:47BA AF          xor          a                ; clear carry
code:47BB 0E 05          ld        c, #5                ; 50 pts
code:47BD
code:47BD          loc_0_47BD:                ; CODE XREF: update_bullet+A2↑j
code:47BD 81          add          a, c                ; multiplier
code:47BE 10 FD          djnz      loc_0_47BD              ; calc ufo score
code:47C0 47          ld        b, a
code:47C1 F5          push       af
code:47C2 CD BB 46    call      update_score_and_chk_bonus_life
code:47C5 F1          pop        af
code:47C6 E1          pop        hl
code:47C7 22 20 40    ld        (0x4020), hl            ; cursor position
code:47CA 06 0A          ld        b, #10
code:47CC 5F          ld        e, a                ; bonus/10
code:47CD 21 00 00    ld        hl, #0
code:47D0 55          ld        d, 1
code:47D1
code:47D1          loc_0_47D1:                ; CODE XREF: update_bullet+B6↑j
code:47D1 19          add          hl, de
code:47D2 10 FD          djnz      loc_0_47D1              ; calc bonus
code:47D4 3E 3C          ld        a, #0x3C ; '<'
code:47D6 CD 33 00    call      0x33                ; display character
code:47D9 CD AF 0F    call      0xFAF                ; display integer in HL
code:47DC 3E 3E          ld        a, #0x3E ; '>'

```

```

code:47DE CD 33 00      call    0x33                ; display character
code:47E1 AF           xor      a
code:47E2 32 10 43      ld      (ufo_active), a      ; flag inactive
code:47E5 32 0C 43      ld      (ufo_timer), a      ; reset timer
code:47E8 18 AC         jr      clear_bullet_active
code:47EA              ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:47EA
code:47EA              chk_and_handle_bullet_hits_bomb:      ; CODE XREF: update_bullet+2E↑j
code:47EA E5           push    hl
code:47EB 21 03 00      ld      hl, #3                ; RAND(1-3)
code:47EE CD 1D 4D      call    rand
code:47F1 7D           ld      a, 1                ; get result
code:47F2 E1           pop     hl
code:47F3 FE 03         cp      #3
code:47F5 F5           push    af
code:47F6 C4 0A 4E      call    NZ, handle_bullet_destroys_bomb
code:47F9 F1           pop     af
code:47FA FE 02         cp      #2
code:47FC DA 38 47      jp      C, display_bullet      ; RAND=1
code:47FF C2 96 47      jp      NZ, clear_bullet_active ; RAND=3
code:4802 36 20         ld      (hl), #0x20 ; ' '      ; display space
code:4804 C3 3A 47      jp      loc_0_473A
code:4807              ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4807
code:4807              loc_0_4807:                        ; CODE XREF: handle_fire+30↑j
code:4807 D9           exx
code:4808 C3 1C 47      jp      update_bullet
code:480B              ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:480B
code:480B              handle_shield_hit:                  ; CODE XREF: update_bullet+3C↑j
code:480B CD 2E 48      call    erode_shield_from_bullet
code:480E 18 86         jr      clear_bullet_active
code:480E              ; End of function update_bullet
code:4810
code:4810              ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000
code:4810
code:4810
code:4810              erode_shield_from_bomb:              ; CODE XREF:
handle_drop_new_bomb+90↑j
code:4810 E5           push    hl                ; update_bombs+96↑p
code:4810              push    bc
code:4811 C5           ld      a, (hl)                ; get character from video
code:4812 7E           ld      c, a
code:4813 4F           ld      a, #0xBC ; '¼'
code:4814 3E BC        and     c
code:4816 A1           cp      c
code:4817 B9           jr      NZ, loc_0_4822
code:4818 20 08        ld      a, #0xB0 ; '°'
code:481A 3E B0        and     c
code:481C A1           cp      c
code:481D B9           jr      NZ, loc_0_4822
code:481E 20 02        ld      a, #0x20 ; ' '
code:4820 3E 20        ld      a, #0x20 ; ' '
code:4822
code:4822              loc_0_4822:                        ; CODE XREF:
erode_shield_from_bomb+8↑j
code:4822 FE 80         cp      #0x80 ; '€'                ; erode_shield_from_bomb+E↑j
code:4824 20 02        jr      NZ, loc_0_4828
code:4826 3E 20        ld      a, #0x20 ; ' '
code:4828
code:4828              loc_0_4828:                        ; CODE XREF:
erode_shield_from_bomb+14↑j
code:4828 77           ld      (hl), a                ; update character
code:4829 C1           pop     bc
code:482A E1           pop     hl
code:482B C3 57 4D      jp      dec_bomb_count
code:482B              ; End of function erode_shield_from_bomb
code:482B
code:482E
code:482E              ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000

```

[illegible]

```

code:4874 ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4874
code:4874 end_of_wave: ; CODE XREF: update_bullet+661j
code:4874 C3 F5 49 jp new_wave
code:4877
code:4877 ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000
code:4877
code:4877
code:4877 check_for_graphic_in_column: ; CODE XREF:
animate_and_move_invaders+C|p
code:4877 11 40 00 ; code:4B8B|p
code:4877 ld de, #64 ; characters/line
code:487A 06 0D ld b, #13
code:487C
code:487C loc_0_487C: ; CODE XREF:
check_for_graphic_in_column+A|j
code:487C CB 7E bit 7, (hl) ; graphic character?
code:487E 20 06 jr NZ, loc_0_4886 ; yes, skip
code:4880 19 add hl, de ; next line
code:4881 10 F9 djnz loc_0_487C ; loop thru 13 lines
code:4883 3E 20 ld a, #0x20 ; ' ' ; flag no match
code:4885 C9 ret
code:4886 ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4886
code:4886 loc_0_4886: ; CODE XREF:
check_for_graphic_in_column+7|j
code:4886 AF xor a ; flag match
code:4887 C9 ret
code:4887 ; End of function check_for_graphic_in_column
code:4887
code:4888
code:4888 ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000
code:4888
code:4888
code:4888 check_and_handle_move: ; CODE XREF: code:4F2B|p
code:4888 3A 08 38 ld a, (0x3808) ; keyboard
code:488B E6 05 and #5 ; "X" or "Z" pressed?
code:488D C8 ret Z ; no return
code:488E FE 04 cp #4 ; "Z"?
code:4890 30 16 jr NC, handle_move_left ; yes, skip
code:4892 3A BB 3F ld a, (video_ram+0x3BB) ; right-most position for base
code:4895 CB 7F bit 7, a ; graphic character?
code:4897 C0 ret NZ ; yes, return (can't move
right)
code:4898 D9 exx
code:4899 21 80 3F ld hl, #video_ram+0x380 ; base row
code:489C CD 3E 46 call move_video_line_right_HL
code:489F 2A 13 43 ld hl, (base_centre)
code:48A2 23 inc hl ; move player right
code:48A3 22 13 43 ld (base_centre), hl
code:48A6
code:48A6 loc_0_48A6: ; CODE XREF:
check_and_handle_move+35|j
code:48A6 D9 exx
code:48A7 C9 ret
code:48A8 ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:48A8
code:48A8 handle_move_left: ; CODE XREF:
check_and_handle_move+8|j
code:48A8 C0 ret NZ
code:48A9 3A 84 3F ld a, (video_ram+0x384) ; left-most position for base
code:48AC CB 7F bit 7, a ; graphic character?
code:48AE C0 ret NZ ; yes, return (can't move left)
code:48AF D9 exx
code:48B0 21 80 3F ld hl, #video_ram+0x380 ; base row
code:48B3 CD 60 46 call move_video_line_left_HL
code:48B6 2A 13 43 ld hl, (base_centre)
code:48B9 2B dec hl ; move player left
code:48BA 22 13 43 ld (base_centre), hl
code:48BD 18 E7 jr loc_0_48A6
code:48BD ; End of function check_and_handle_move
code:48BD

```



```

code:4913          loc_0_4913:                                ; CODE XREF: update_ufo+12↑j
code:4913 36 BB          ld      (hl), #0xBB ; '»'          ; display
code:4915 18 F3          jr      loc_0_490A
code:4917          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4917
code:4917          move_ufo:                                  ; CODE XREF: update_ufo+17↑j
code:4917 3A 0D 43        ld      a, (ufo_dir)
code:491A B7            or      a
code:491B 28 08          jr      Z, move_ufo_left          ; left?
code:491D 21 00 3C        ld      hl, #video_ram
code:4920 CD 3E 46        call   move_video_line_right_HL
code:4923 18 06          jr      ufo_TTL_tick
code:4925          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4925
code:4925          move_ufo_left:                              ; CODE XREF: update_ufo+25↑j
code:4925 21 00 3C        ld      hl, #video_ram
code:4928 CD 60 46        call   move_video_line_left_HL
code:492B
code:492B          ufo_TTL_tick:                              ; CODE XREF: update_ufo+2D↑j
code:492B 3A 0B 43        ld      a, (ufo_TTL)
code:492E 3D            dec     a
code:492F 32 0B 43        ld      (ufo_TTL), a
code:4932 28 02          jr      Z, flag_ufo_inactive      ; ufo still active?
code:4934          ; no, skip
code:4934          loc_0_4934:                                  ; CODE XREF: update_ufo+44↑j
code:4934 D9            exx
code:4935 C9            ret
code:4936          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4936
code:4936          flag_ufo_inactive:                          ; CODE XREF: update_ufo+3C↑j
code:4936 AF            xor     a
code:4937 32 10 43        ld      (ufo_active), a
code:493A 18 F8          jr      loc_0_4934
code:493A          ; End of function update_ufo
code:493A
code:493C          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:493C
code:493C          game_over:                                  ; CODE XREF: code:4AB4↑j
code:493C 31 8A 42        ; code:4B45↑j
code:493C          ld      sp, #0x428A
code:493F 21 19 3C        ld      hl, #video_ram+0x19
code:4942 22 20 40        ld      (0x4020), hl
code:4945 21 00 3C        ld      hl, #video_ram
code:4948 CD 7E 4C        call   clear_video_line_HL
code:494B 21 93 44        ld      hl, #aGAMEOVER
code:494E CD 05 45        call   display_message_slowly
code:4951 CD 5F 4C        call   check_for_new_high_score
code:4954 01 FF FF        ld      bc, #65535
code:4957 CD 60 00        call   0x60
code:495A CD 60 00        call   0x60
code:495D CD 60 00        call   0x60
code:4960 C3 4F 45        jp      attract_loop
code:4963
code:4963          ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000
code:4963
code:4963
code:4963          display_invader_row:                          ; CODE XREF: code:4A78↑p
code:4963 E5          ; code:4A81↑p ...
code:4963          push     hl
code:4964 D5          push     de
code:4965 C5          push     bc
code:4966 06 0A        ld      b, #10
code:4968 D5          push     de
code:4969 11 40 00        ld      de, #64
code:496C ED 52        sbc     hl, de
code:496E D1          pop      de
code:496F 23          inc     hl
code:4970
code:4970          loc_0_4970:                                  ; CODE XREF:
display_invader_row+1C↑j
code:4970 22 20 40        ld      (0x4020), hl
code:4973 D5          push     de

```


[illegible]

```

check_for_R_key+5↑j
code:49C2 AF          xor      a
code:49C3 32 0C 43    ld        (ufo_timer), a
code:49C6 32 15 43    ld        (unused_4315), a
code:49C9 32 12 43    ld        (unused_4312), a
code:49CC 32 0D 43    ld        (ufo_dir), a
code:49CF 32 0E 43    ld        (wave_no), a
code:49D2 32 10 43    ld        (ufo_active), a
code:49D5 32 11 43    ld        (bullet_active), a
code:49D8 32 22 43    ld        (keybd_state), a
code:49DB 3C          inc      a
code:49DC 32 0A 43    ld        (invader_dir), a
code:49DF 3E 05       ld        a, #5
code:49E1 32 0C 43    ld        (ufo_timer), a
code:49E4 3E 03       ld        a, #3
code:49E6 32 0F 43    ld        (no_lives), a
code:49E9 CD 90 4E     call     display_lives_left
code:49EC 31 8A 42     ld        sp, #0x428A
code:49EF CD B0 46     call     zero_score
code:49F2 CD 31 4E     call     display_GOOD_LUCK
code:49F5
code:49F5          new_wave:                                ; CODE XREF: code:4874↑j
code:49F5 31 8A 42     ld        sp, #0x428A
code:49F8 AF          xor      a
code:49F9 32 11 43    ld        (bullet_active), a        ; clear fired flag
code:49FC 3A 0E 43    ld        a, (wave_no)
code:49FF 3C          inc      a                ; next wave number
code:4A00 FE 07       cp        #7                ; highest?
code:4A02 20 02       jr        NZ, loc_0_4A06        ; no, skip
code:4A04 3E 01       ld        a, #1                ; reset to 1
code:4A06
code:4A06          loc_0_4A06:                                ; CODE XREF: code:4A02↑j
code:4A06 32 0E 43    ld        (wave_no), a
code:4A09 E6 06       and      #6                ; 2/4/6
code:4A0B 11 40 00     ld        de, #0x40 ; '@'        ; characters/line
code:4A0E 62          ld        h, d
code:4A0F 6B          ld        l, e                ; hl=0x0040
code:4A10 06 01       ld        b, #1
code:4A12 FE 02       cp        #2                ; compare wave_no with 2
code:4A14 38 06       jr        C, loc_0_4A1C        ; wave_no=1, skip
code:4A16 28 02       jr        Z, loc_0_4A1A        ; wave_no=2, skip
code:4A18 04          inc      b
code:4A19 19          add      hl, de
code:4A1A
code:4A1A          loc_0_4A1A:                                ; CODE XREF: code:4A16↑j
code:4A1A 04          inc      b
code:4A1B 19          add      hl, de
code:4A1C
code:4A1C          loc_0_4A1C:                                ; CODE XREF: code:4A14↑j
code:4A1C EB          ex        de, hl
code:4A1D
code:4A1D          calc_invader_row_addr:                            ; CODE XREF: code:4A40↑j
code:4A1D C5          push     bc
code:4A1E DD 21 16 43  ld        ix, #init_row_1_invader_addr
code:4A22 FD 21 02 43  ld        iy, #row_1_invader_addr
code:4A26 06 04       ld        b, #4                ; 4 rows of invaders
code:4A28
code:4A28          loc_0_4A28:                                ; CODE XREF: code:4A3D↑j
code:4A28 DD 6E 00     ld        l, 0(ix)
code:4A2B DD 66 01     ld        h, 1(ix)
code:4A2E 19          add      hl, de                ; calc video address for
invader row
code:4A2F FD 75 00     ld        0(iy), l
code:4A32 FD 74 01     ld        1(iy), h                ; store
code:4A35 DD 23       inc      ix
code:4A37 DD 23       inc      ix                ; next row address
code:4A39 FD 23       inc      iy
code:4A3B FD 23       inc      iy
code:4A3D 10 E9       djnz     loc_0_4A28        ; loop thru all rows of
invaders
code:4A3F C1          pop      bc
code:4A40 10 DB       djnz     calc_invader_row_addr
code:4A42 CD 18 45     call     wipe_screen_left_to_right_slow
code:4A45 21 09 3F     ld        hl, #video_ram+0x309    ; cursor position
code:4A48 22 20 40     ld        (0x4020), hl
code:4A4B 21 51 43     ld        hl, #shield            ; shield #1
code:4A4E E5          push     hl

```

```

code:4A4F CD E5 44      call    display_message
code:4A52 21 17 3F      ld      hl, #video_ram+0x317      ; cursor position
code:4A55 22 20 40      ld      hl, (0x4020), hl
code:4A58 E1           pop     hl
code:4A59 E5           push    hl                      ; shield #2
code:4A5A CD E5 44      call    display_message
code:4A5D 21 24 3F      ld      hl, #video_ram+0x324      ; cursor position
code:4A60 22 20 40      ld      hl, (0x4020), hl
code:4A63 E1           pop     hl
code:4A64 E5           push    hl                      ; shield #3
code:4A65 CD E5 44      call    display_message
code:4A68 21 31 3F      ld      hl, #video_ram+0x331      ; cursor position
code:4A6B 22 20 40      ld      hl, (0x4020), hl
code:4A6E E1           pop     hl                      ; shield #4
code:4A6F CD E5 44      call    display_message
code:4A72 11 23 43      ld      de, #invader_30pt
code:4A75 2A 02 43      ld      hl, (row_1_invader_addr)
code:4A78 CD 63 49      call    display_invader_row      ; display 1st row of invaders
code:4A7B 11 30 43      ld      de, #invader_20pt
code:4A7E 2A 04 43      ld      hl, (row_2_invader_addr)
code:4A81 CD 63 49      call    display_invader_row      ; display 2nd row of invaders
code:4A84 11 3D 43      ld      de, #invader_10pt
code:4A87 2A 06 43      ld      hl, (row_3_invader_addr)
code:4A8A CD 63 49      call    display_invader_row      ; display 3rd row of invaders
code:4A8D 2A 08 43      ld      hl, (row_4_invader_addr)
code:4A90 CD 63 49      call    display_invader_row      ; display 4th row of invaders
code:4A93 3E 28         ld      a, #40                      ; number of invaders left
code:4A95 32 01 43      ld      (invaders_left), a
code:4A98 32 21 43      ld      (invader_timer), a
code:4A9B
code:4A9B      init_and_display_player_base:      ; CODE XREF: code:4AD9|j
code:4A9B      ld      hl, #video_ram+0x384      ; cursor position
code:4A9E 22 20 40      ld      hl, (0x4020), hl
code:4AA1 21 64 43      ld      hl, #player
code:4AA4 CD E5 44      call    display_message      ; draw player base
code:4AA7 21 86 3F      ld      hl, #video_ram+0x386
code:4AAA 22 13 43      ld      (base_centre), hl      ; center of base
code:4AAD C3 FB 4E      jp      init_turn
code:4AB0
code:4AB0      ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4AB0
code:4AB0      decrement_player_life:      ; CODE XREF: code:4B2A|j
code:4AB0 3A 0F 43      ld      a, (no_lives)
code:4AB3 3D         dec     a                      ; any lives left?
code:4AB4 CA 3C 49      jp      Z, game_over      ; no, exit
code:4AB7 32 0F 43      ld      (no_lives), a
code:4ABA CD 90 4E      call    display_lives_left
code:4ABD AF         xor     a
code:4ABE 32 11 43      ld      (bullet_active), a      ; clear fired flag
code:4AC1 32 12 43      ld      (unused_4312), a
code:4AC4 CD 3B 4D      call    restore_space_characters
code:4AC7 21 80 3F      ld      hl, #video_ram+0x380
code:4ACA CD 7E 4C      call    clear_video_line_HL
code:4ACD 01 FF FF      ld      bc, #65535      ; ~1s
code:4AD0 CD 60 00      call    0x60      ; delay
code:4AD3 CD 60 00      call    0x60      ; delay
code:4AD6 31 8A 42      ld      sp, #0x428A
code:4AD9 C3 9B 4A      jp      init_and_display_player_base
code:4ADC
code:4ADC      ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000000000000000000
code:4ADC
code:4ADC      invert_display:      ; CODE XREF:
animate_player_hit+4|p
code:4ADC 21 00 3C      ld      hl, #video_ram      ; animate_player_hit+B|j ...
code:4ADC      ld      hl, #video_ram
code:4ADF 01 00 04      ld      bc, #0x400      ; video ram size
code:4AE2
code:4AE2      loc_0_4AE2:      ; CODE XREF:
invert_display+1C|j
code:4AE2 7E         ld      a, (hl)      ; get character
code:4AE3 FE 20      cp      #0x20 ; ' '      ; space?
code:4AE5 20 02      jr      NZ, loc_0_4AE9      ; no, skip
code:4AE7 36 80      ld      (hl), #0x80 ; '€'      ; graphic space
code:4AE9
code:4AE9      loc_0_4AE9:      ; CODE XREF: invert_display+9|j

```

```

code:4AE9 CB 7E          bit      7, (hl)                ; graphics character?
code:4AEB 28 07          jr       Z, loc_0_4AF4           ; no, skip
code:4AED 7E            ld       a, (hl)                 ; get character
code:4AEE 2F            cpl              ; invert
code:4AEF CB FF          set       7, a                   ; make graphics character
code:4AF1 CB B7          res       6, a                   ; 1st block of graphics
characters
code:4AF3 77            ld        (hl), a                 ; display
code:4AF4               loc_0_4AF4:                       ; CODE XREF: invert_display+F↑j
code:4AF4 23            inc        hl                     ; next video address
code:4AF5 0B            dec        bc
code:4AF6 78            ld         a, b
code:4AF7 B1            or         c
code:4AF8 20 E8          jr       NZ, loc_0_4AE2           ; loop through screen
code:4AFA C9            ret
code:4AFA               ; End of function invert_display
code:4AFB
code:4AFB               ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000
code:4AFB
code:4AFB
code:4AFB             get_player_address:                  ; CODE XREF:
animate_player_hit|p
code:4AFB 2A 13 43       ld        hl, (base_centre)
code:4AFE 2B            dec        hl
code:4AFF 2B            dec        hl
code:4B00 C9            ret
code:4B00               ; End of function get_player_address
code:4B01
code:4B01               ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000
code:4B01
code:4B01
code:4B01             animate_base_hit:                    ; CODE XREF:
animate_player_hit+8|p
code:4B01 3E A6          ld        a, #0xA6 ; '|'          ; hash graphic
code:4B03 06 00          ld        b, #0                ; 256 times
code:4B05
code:4B05             loc_0_4B05:                          ; CODE XREF:
animate_base_hit+16|j
code:4B05 C5            push     bc
code:4B06 E5            push     hl                    ; player base address
code:4B07 06 05          ld        b, #5                ; 5 chars to display
code:4B09 EE 3F          xor       #0x3F ; '?'          ; invert hash
code:4B0B
code:4B0B             loc_0_4B0B:                          ; CODE XREF:
animate_base_hit+C|j
code:4B0B 77            ld        (hl), a                ; display hash
code:4B0C 23            inc        hl                    ; next video address
code:4B0D 10 FC          djnz     loc_0_4B0B           ; loop through 5 chars
code:4B0F
code:4B0F             loc_0_4B0F:                          ; CODE XREF:
animate_base_hit+12|j
code:4B0F E3            ex        (sp), hl
code:4B10 E3            ex        (sp), hl
code:4B11 E3            ex        (sp), hl
code:4B12 E3            ex        (sp), hl            ; delay
code:4B13 10 FA          djnz     loc_0_4B0F           ; loop 256 times
code:4B15 E1            pop       hl
code:4B16 C1            pop       bc
code:4B17 10 EC          djnz     loc_0_4B05           ; loop 256 times
code:4B19 C9            ret
code:4B19               ; End of function animate_base_hit
code:4B19
code:4B1A
code:4B1A               ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000
code:4B1A
code:4B1A
code:4B1A             animate_player_hit:                  ; CODE XREF: code:4B27|p
code:4B1A CD FB 4A       call     get_player_address
code:4B1D E5            push     hl
code:4B1E CD DC 4A       call     invert_display
code:4B21 E1            pop       hl

```

```

code:4B22 CD 01 4B          call    animate_base_hit
code:4B25 18 B5             jr      invert_display
code:4B25                   ; End of function animate_player_hit
code:4B25
code:4B27                   ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4B27
code:4B27                   handle_base_hit:                               ; CODE XREF: update_bombs+93|j
code:4B27 CD 1A 4B          call    animate_player_hit
code:4B2A C3 B0 4A          jp      decrement_player_life
code:4B2D                   ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4B2D
code:4B2D                   flash_screen_10_times:                         ; CODE XREF: code:4C02|j
code:4B2D 06 0A             ld      b, #10                               ; 10 times
code:4B2F
code:4B2F                   flash_screen:                                   ; CODE XREF: code:4B43|j
code:4B2F C5               push    bc
code:4B30 CD DC 4A          call    invert_display
code:4B33 01 10 27          ld      bc, #10000                               ; ~140ms
code:4B36 CD 60 00          call    0x60                                           ; delay
code:4B39 CD DC 4A          call    invert_display
code:4B3C 01 10 27          ld      bc, #10000                               ; ~140ms
code:4B3F CD 60 00          call    0x60                                           ; delay
code:4B42 C1               pop     bc
code:4B43 10 EA             djnz    flash_screen                               ; repeat
code:4B45 C3 3C 49          jp      game_over
code:4B48
code:4B48                   ; 0000000000000000 S U B R O U T I N E
000000000000000000000000000000000000000000000000000000000000
code:4B48
code:4B48
code:4B48                   animate_and_move_invaders:                       ; CODE XREF: code:4F71|p
code:4B48 E5               push    hl
code:4B49 D5               push    de
code:4B4A C5               push    bc
code:4B4B 3A 0A 43          ld      a, (invader_dir)
code:4B4E B7               or      a                                           ; left?
code:4B4F 28 37             jr      Z, animate_and_move_invaders_left ; yes, skip
code:4B51 21 7F 3C          ld      hl, #video_ram+0x7F                         ; end of 2nd line on screen
code:4B54 CD 77 48          call    check_for_graphic_in_column
code:4B57 B7               or      a                                           ; invaders reached RHS of
screen?
code:4B58 CA BE 4B          jp      Z, set_invader_dir_left ; yes, skip
code:4B5B 2A 08 43          ld      hl, (row_4_invader_addr)
code:4B5E CD 7A 4B          call    move_invader_row_right
code:4B61 2A 06 43          ld      hl, (row_3_invader_addr)
code:4B64 CD 7A 4B          call    move_invader_row_right
code:4B67 2A 04 43          ld      hl, (row_2_invader_addr)
code:4B6A CD 7A 4B          call    move_invader_row_right
code:4B6D 2A 02 43          ld      hl, (row_1_invader_addr)
code:4B70 CD 7A 4B          call    move_invader_row_right
code:4B73 CD FC 45          call    animate_invaders
code:4B76
code:4B76                   move_invaders_down_ret:                       ; CODE XREF: code:4BAC|j
code:4B76 C1               pop     bc                                           ; code:4C1D|j ...
code:4B77 D1               pop     de
code:4B78 E1               pop     hl
code:4B79 C9               ret
code:4B79                   ; End of function animate_and_move_invaders
code:4B7A
code:4B7A                   ; 0000000000000000 S U B R O U T I N E
000000000000000000000000000000000000000000000000000000000000
code:4B7A
code:4B7A
code:4B7A                   move_invader_row_right:                       ; CODE XREF:
animate_and_move_invaders+16|p
code:4B7A 7C               ;
animate_and_move_invaders+1C|p ...
code:4B7A                   ld      a, h
code:4B7B B7               or      a                                           ; any invaders left on this
row?
code:4B7C C8               ret      Z                                           ; no, return
code:4B7D CD 3E 46          call    move_video_line_right_HL
code:4B80 11 40 00          ld      de, #64                               ; characters/line

```

```

code:4B83 ED 52          sbc      hl, de          ; line above
code:4B85 C3 3E 46          jp      move_video_line_right HL
code:4B85          ; End of function move_invader_row_right
code:4B85
code:4B88          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4B88
code:4B88          animate_and_move_invaders_left:          ; CODE XREF:
animate_and_move_invaders+7↑j
code:4B88 21 40 3C          ld      hl, #video_ram+0x40      ; start of 2nd line
code:4B8B 2D 77 48          call   check_for_graphic_in_column
code:4B8E B7              or      a          ; invaders reached LHS of
screen?
code:4B8F 28 38          jr      Z, move_invaders_down      ; yes, skip
code:4B91 CD FC 45          call   animate_invaders
code:4B94
code:4B94          move_invaders_left:          ; CODE XREF: code:4BC6↑j
code:4B94 2A 08 43          ld      hl, (row_4_invader_addr)
code:4B97 CD AE 4B          call   move_invader_row_left
code:4B9A 2A 06 43          ld      hl, (row_3_invader_addr)
code:4B9D CD AE 4B          call   move_invader_row_left
code:4BA0 2A 04 43          ld      hl, (row_2_invader_addr)
code:4BA3 CD AE 4B          call   move_invader_row_left
code:4BA6 2A 02 43          ld      hl, (row_1_invader_addr)
code:4BA9 CD AE 4B          call   move_invader_row_left
code:4BAC 18 C8          jr      move_invaders_down_ret
code:4BAE
code:4BAE          ; 0000000000000000 S U B R O U T I N E
000000000000000000000000000000000000000000000000000000000000
code:4BAE
code:4BAE
code:4BAE          move_invader_row_left:          ; CODE XREF: code:4B97↑p
code:4BAE 7C              ; code:4B9D↑p ...
code:4BAE          ld      a, h
code:4BAF B7              or      a          ; any invaders left on row?
code:4BB0 C8              ret      Z          ; no, return
code:4BB1 E5              push   hl
code:4BB2 CD 60 46          call   move_video_line_left_HL
code:4BB5 E1              pop      hl
code:4BB6 11 40 00          ld      de, #64          ; characters/line
code:4BB9 ED 52          sbc      hl, de          ; line above
code:4BBB C3 60 46          jp      move_video_line_left_HL
code:4BBB          ; End of function move_invader_row_left
code:4BBB
code:4BBE          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4BBE
code:4BBE          set_invader_dir_left:          ; CODE XREF:
animate_and_move_invaders+10↑j
code:4BBE 3A 0A 43          ld      a, (invader_dir)
code:4BC1 EE 01          xor      #1          ; toggle invader direction
code:4BC3 32 0A 43          ld      (invader_dir), a
code:4BC6 C3 94 4B          jp      move_invaders_left
code:4BC9          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4BC9
code:4BC9          move_invaders_down:          ; CODE XREF: code:4B8F↑j
code:4BC9 3A 11 43          ld      a, (bullet_active)
code:4BCC B7              or      a          ; fired?
code:4BCD 28 05          jr      Z, loc_0_4BD4      ; no, skip
code:4BCF 2A 1E 43          ld      hl, (bullet_addr)
code:4BD2 36 20          ld      (hl), #0x20; ' '          ; display space
code:4BD4
code:4BD4          loc_0_4BD4:          ; CODE XREF: code:4BCD↑j
code:4BD4 DD 21 08 43          ld      ix, #row_4_invader_addr
code:4BD8 06 04          ld      b, #4          ; 4 rows to check
code:4BDA
code:4BDA          loc_0_4BDA:          ; CODE XREF: code:4BE9↑j
code:4BDA DD 6E 00          ld      l, 0(ix)
code:4BDD DD 66 01          ld      h, 1(ix)          ; hl = invader addr
code:4BE0 7C              ld      a, h
code:4BE1 B7              or      a          ; any invaders left on this
line?
code:4BE2 C4 2E 4C          call   NZ, move_invader_row_down ; yes, call
code:4BE5 DD 2B          dec      ix
code:4BE7 DD 2B          dec      ix          ; next invader row address
code:4BE9 10 EF          djnz   loc_0_4BDA          ; loop thru 4 rows of invaders

```



```
00000000000000000000000000000000000000000000
code:4C55
code:4C55
code:4C55          get_video_line_below_invaders:                ; CODE XREF: code:4BFA↑p
code:4C55 7C            ld      a, h
code:4C56 B7            or       a
                        ; any invaders left in row?
code:4C57 C8            ret      Z
                        ; no, return
code:4C58 D5            push    de
code:4C59 11 40 00     ld      de, #64
                        ; characters/line
code:4C5C 19            add     hl, de
                        ; next line
code:4C5D D1            pop     de
code:4C5E C9            ret
code:4C5E          ; End of function get_video_line_below_invaders
code:4C5E
code:4C5F
code:4C5F          ; 000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000
code:4C5F
code:4C5F
code:4C5F          check_for_new_high_score:                      ; CODE XREF: code:4951↑p
code:4C5F 21 C9 3F      ld      hl, #video_ram+0x3C9
                        ; score
code:4C62 11 FA 3F      ld      de, #video_ram+0x3FA
                        ; high score
code:4C65 06 04         ld      b, #4
                        ; 4 digits to compare
code:4C67
code:4C67          loc_0_4C67:                                      ; CODE XREF:
check_for_new_high_score+10↑j
code:4C67 4E            ld      c, (hl)
                        ; get score digit
code:4C68 1A            ld      a, (de)
                        ; get high score digit
code:4C69 B9            cp      c
                        ; score higher?
code:4C6A 38 06        jr      C, update_high_score
                        ; yes, skip
code:4C6C C0            ret     NZ
                        ; done if not the same
code:4C6D 23            inc     hl
code:4C6E 13            inc     de
                        ; next digits
code:4C6F 10 F6        djnz   loc_0_4C67
                        ; loop through all digits
code:4C71 C9            ret
code:4C72
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4C72
code:4C72          update_high_score:                              ; CODE XREF:
check_for_new_high_score+B↑j
code:4C72 21 C9 3F      ld      hl, #video_ram+0x3C9
                        ; source = score
code:4C75 11 FA 3F      ld      de, #video_ram+0x3FA
                        ; destination = high score
code:4C78 01 05 00     ld      bc, #5
                        ; 5 digits to copy
code:4C7B ED B0        ldir
                        ; copy
code:4C7D C9            ret
code:4C7D          ; End of function check_for_new_high_score
code:4C7D
code:4C7E
code:4C7E          ; 000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000
code:4C7E
code:4C7E
code:4C7E          clear_video_line_HL:                            ; CODE XREF: update_bullet+94↑p
code:4C7E C5            push    bc
                        ; code:4948↑p ...
code:4C7F D5            push    de
code:4C80 06 40        ld      b, #64
                        ; characters/line
code:4C82
code:4C82          loc_0_4C82:                                      ; CODE XREF:
clear_video_line_HL+7↑j
code:4C82 36 20        ld      (hl), #0x20 ; ' '
                        ; display space
code:4C84 23            inc     hl
                        ; next video address
code:4C85 10 FB        djnz   loc_0_4C82
                        ; clear a line
code:4C87 D1            pop     de
code:4C88 C1            pop     bc
code:4C89 C9            ret
code:4C89          ; End of function clear_video_line_HL
code:4C89
code:4C8A
code:4C8A          ; 000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000
code:4C8A
code:4C8A
code:4C8A          handle_drop_new_bomb:                          ; CODE XREF: code:4F65↑p
code:4C8A D9            exx
code:4C8B 06 04        ld      b, #4
                        ; number of invader rows
code:4C8D DD 21 08 43  ld      ix, #row 4 invader addr
```



```

code:4C91
code:4C91          loc_0_4C91:                      ; CODE XREF:
handle_drop_new_bomb+11↑j
code:4C91 DD 7E 01          ld      a, 1(ix)
code:4C94 B7              or      a                      ; any invaders left on this
row?
code:4C95 20 08          jr      NZ, check_and_handle_new_bomb ; yes, continue
code:4C97 DD 2B          dec     ix
code:4C99 DD 2B          dec     ix                      ; next row above
code:4C9B 10 F4          djnz    loc_0_4C91              ; loop thru all rows
code:4C9D
code:4C9D          init_bomb_ret:                    ; CODE XREF:
handle_drop_new_bomb+3B↑j
code:4C9D D9              ; handle_drop_new_bomb+4E↑j ...
code:4C9D          exx
code:4C9E C9              ret
code:4C9F
code:4C9F          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4C9F
code:4C9F          check_and_handle_new_bomb:        ; CODE XREF:
handle_drop_new_bomb+B↑j
code:4C9F 2A 13 43          ld      hl, (base_centre)
code:4CA2 11 80 C0          ld      de, #0xC080
code:4CA5 19              add     hl, de                      ; base X position
code:4CA6 EB              ex      de, hl                      ; base X position
code:4CA7 DD 6E 00          ld      l, 0(ix)
code:4CAA DD 66 01          ld      h, 1(ix)                      ; invader row address
code:4CAD E5              push    hl
code:4CAE 21 03 00          ld      hl, #3                      ; RAND(1-3)
code:4CB1 CD 1D 4D          call    rand
code:4CB4 7D              ld      a, 1                      ; get result
code:4CB5 E1              pop     hl                      ; invader row address
code:4CB6 FE 01          cp      #1                      ; drop a bomb near the base?
code:4CB8 20 4B          jr      NZ, random_bomb_x_position ; no, random
code:4CBA
code:4CBA          loc_0_4CBA:                        ; CODE XREF:
handle_drop_new_bomb+84↑j
code:4CBA 19              add     hl, de                      ; calc bomb X position
code:4CBB 11 80 FF          ld      de, #0xFF80              ; offset of 2 video lines above
code:4CBE
code:4CBE          loc_0_4CBE:                        ; CODE XREF:
handle_drop_new_bomb+39↑j
code:4CBE CB 7E          bit     7, (hl)                  ; invader above bomb position?
code:4CC0 20 05          jr      NZ, init_new_bomb          ; yes, continue
code:4CC2 19              add     hl, de                      ; 2 lines above
code:4CC3 10 F9          djnz    loc_0_4CBE              ; find invader above
code:4CC5 18 D6          jr      init_bomb_ret            ; no invaders, no bomb
code:4CC7
code:4CC7          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4CC7
code:4CC7          init_new_bomb:                      ; CODE XREF:
handle_drop_new_bomb+36↑j
code:4CC7 DD 21 EA 4E          ld      ix, #bomb_tbl
code:4CCB 06 04          ld      b, #4
code:4CCD
code:4CCD          find_free_bomb_entry:                ; CODE XREF:
handle_drop_new_bomb+4C↑j
code:4CCD DD 7E 01          ld      a, 1(ix)
code:4CD0 B7              or      a                      ; bomb active?
code:4CD1 28 08          jr      Z, loc_0_4CDB              ; no, continue
code:4CD3 CD 2A 4E          call    add_3_to_ix              ; next table location
code:4CD6 10 F5          djnz    find_free_bomb_entry
code:4CD8 C3 9D 4C          jp      init_bomb_ret            ; no free entries, return
code:4CDB
code:4CDB          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4CDB
code:4CDB          loc_0_4CDB:                        ; CODE XREF:
handle_drop_new_bomb+47↑j
code:4CDB CD 28 4D          call    get_invader_address
code:4CDE 11 82 00          ld      de, #0x82 ; ',', '      ; 2 lines below and 2 chars
right
code:4CE1 19              add     hl, de                      ; centre under invader
code:4CE2 E5              push    hl
code:4CE3 21 03 00          ld      hl, #3                      ; rand(1-3)
code:4CE6 CD 1D 4D          call    rand
code:4CE9 11 D2 4E          ld      de, #base_icon+3          ; bomb_characters
code:4CEC 19              add     hl, de                      ; get random character

```

```

code:4CED D1                pop    de                ; centre under invader
code:4CEE CB 7B             bit     7, e
code:4CF0 28 1E             jr      Z, check_new_bomb_shield
code:4CF2
code:4CF2                init_bomb_entry:                ; CODE XREF:
handle_drop_new_bomb+8A|j
code:4CF2 1A                ld      a, (de)                ; get character from video
code:4CF3 FE 20             cp      #0x20 ; ' '                ; space?
code:4CF5 20 A6             jr      NZ, init_bomb_ret                ; no, exit
code:4CF7 7E                ld      a, (hl)                ; get bomb character from table
code:4CF8 12                ld      (de), a                ; display
code:4CF9 DD 73 00          ld      0(ix), e
code:4CFC DD 72 01          ld      1(ix), d                ; store bomb address
code:4CFF DD 77 02          ld      2(ix), a                ; store bomb character
code:4D02 C3 9D 4C          jp      init_bomb_ret
code:4D05                ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4D05
code:4D05                random_bomb_x_position:                ; CODE XREF:
handle_drop_new_bomb+2E|j
code:4D05 E5                push   hl
code:4D06 21 40 00          ld      hl, #64                ; RAND(1-64)
code:4D09 CD 1D 4D          call   rand
code:4D0C EB                ex      de, hl                ; DE = result
code:4D0D E1                pop     hl
code:4D0E 18 AA            jr      loc_0_4CBA
code:4D10                ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4D10
code:4D10                check_new_bomb_shield:                ; CODE XREF:
handle_drop_new_bomb+66|j
code:4D10 EB                ex      de, hl                ; HL = centre under invader
code:4D11 CB 7E             bit     7, (hl)                ; graphic character?
code:4D13 EB                ex      de, hl
code:4D14 28 DC            jr      Z, init_bomb_entry                ; no, continue
code:4D16 EB                ex      de, hl
code:4D17 E5                push   hl
code:4D18 D9                exx
code:4D19 E1                pop     hl
code:4D1A C3 10 48          jp      erode_shield_from_bomb
code:4D1A                ; End of function handle_drop_new_bomb
code:4D1A
code:4D1D
code:4D1D                ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000
code:4D1D
code:4D1D
code:4D1D                rand:                ; CODE XREF: update_bullet+9A|p
code:4D1D D5                ; update_bullet+D2|p ...
code:4D1D                push   de
code:4D1E C5                push   bc
code:4D1F CD CC 14          call   0x14CC                ; ROM RAND() function
code:4D22 CD 7F 0A          call   0xA7F                ; transfer result to HL
code:4D25 C1                pop     bc
code:4D26 D1                pop     de
code:4D27 C9                ret
code:4D27                ; End of function rand
code:4D27
code:4D28
code:4D28                ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000
code:4D28
code:4D28                get_invader_address:                ; CODE XREF: update_bullet+3F|p
code:4D28 D5                ; handle_drop_new_bomb+51|p
code:4D28                push   de
code:4D29 7E                ld      a, (hl)                ; character at video address
code:4D2A E6 30            and     #0x30 ; '0'                ; any pixels on bottom row of
cell?
code:4D2C 11 C0 FF          ld      de, #0xFFC0                ; offset of line above
code:4D2F 20 01            jr      NZ, loc_0_4D32                ; yes, skip (top half of
invader)
code:4D31 19                add     hl, de                ; line above (top half of
invader)
code:4D32
code:4D32                loc_0_4D32:                ; CODE XREF:
get_invader_address+7|j

```

[illegible]

```
code:4D6C          call    add_3_to_ix          ; next bomb entry
code:4D6F 10 F5     djnz    loc_0_4D66          ; loop thru all bombs
code:4D71 D9        exx
code:4D72 C9        ret
code:4D73          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4D73          ;
code:4D73          check_and_update_bomb:        ; CODE XREF: update_bombs+B↑j
code:4D73 DD 6E 00  ld      l, 0(ix)
code:4D76 DD 66 01  ld      h, 1(ix)          ; bomb address
code:4D79 DD 7E 02  ld      a, 2(ix)          ; bomb character
code:4D7C BE        cp      (hl)          ; same character?
code:4D7D 28 05     jr      Z, update_bomb      ; yes, continue
code:4D7F CD 53 4D  call    delete_bomb
code:4D82 18 E8     jr      next_bomb
code:4D84          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4D84          ;
code:4D84          update_bomb:                  ; CODE XREF: update_bombs+1E↑j
code:4D84 36 20     ld      (hl), #0x20 ; ' '      ; display space
code:4D86 11 40 00  ld      de, #64          ; characters/line
code:4D89 19        add     hl, de          ; next line down
code:4D8A DD 75 00  ld      0(ix), l
code:4D8D DD 74 01  ld      l, 1(ix), h          ; update bomb address
code:4D90 E5        push    hl
code:4D91 11 C0 3F  ld      de, #video_ram+0x3C0 ; bottom line of video
code:4D94 CD E2 45  call    compare_video_addresses
code:4D97 B7        or      a          ; reached bottom of screen?
code:4D98 F5        push    af
code:4D99 20 2D     jr      NZ, delete_bomb_and_loop ; yes, delete bomb
code:4D9B F1        pop     af
code:4D9C E1        pop     hl          ; bomb address
code:4D9D 7E        ld      a, (hl)          ; get character from video
code:4D9E FE 5B     cp      #0x5B ; '['          ; player bullet?
code:4DA0 28 0A     jr      Z, handle_bullet_hit_bomb ; yes, skip
code:4DA2 FE 81     cp      #0x81 ; '['          ; graphic (non-blank)?
code:4DA4 30 35     jr      NC, check_and_handle_bomb_hit ; yes, skip
code:4DA6 DD 7E 02  ld      a, 2(ix)          ; bomb character
code:4DA9 77        ld      (hl), a          ; display
code:4DAA 18 C0     jr      next_bomb
code:4DAC          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4DAC          ;
code:4DAC          handle_bullet_hit_bomb:        ; CODE XREF: update_bombs+41↑j
code:4DAC E5        push    hl
code:4DAD F5        push    af
code:4DAE 21 03 00  ld      hl, #3
code:4DB1 CD 1D 4D  call    rand
code:4DB4 7D        ld      a, l
code:4DB5 FE 02     cp      #2
code:4DB7 38 0F     jr      C, delete_bomb_and_loop
code:4DB9 20 14     jr      NZ, handle_bomb_destroys_bullet
code:4DBB AF        xor     a
code:4DBC 32 11 43  ld      (bullet_active), a
code:4DBF CD 53 4D  call    delete_bomb          ; both destroyed
code:4DC2 F1        pop     af
code:4DC3 E1        pop     hl
code:4DC4 36 20     ld      (hl), #0x20 ; ' '
code:4DC6 18 A4     jr      next_bomb
code:4DC8          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4DC8          ;
code:4DC8          delete_bomb_and_loop:          ; CODE XREF: update_bombs+3A↑j
code:4DC8 CD 53 4D  ; update_bombs+58↑j ...
code:4DC8          call    delete_bomb
code:4DCB F1        pop     af
code:4DCC E1        pop     hl
code:4DCD 18 9D     jr      next_bomb
code:4DCF          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4DCF          ;
code:4DCF          handle_bomb_destroys_bullet:    ; CODE XREF: update_bombs+5A↑j
code:4DCF AF        xor     a
code:4DD0 32 11 43  ld      (bullet_active), a      ; flag inactive
code:4DD3 F1        pop     af
code:4DD4 E1        pop     hl
code:4DD5 DD 7E 02  ld      a, 2(ix)          ; bomb character
```

```

code:4DD8 77          ld      (hl), a          ; display
code:4DD9 18 91       jr      next_bomb
code:4DDB           ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4DDB
code:4DDB          check_and_handle_bomb_hit:          ; CODE XREF: update_bombs+45↑j
code:4DDC EB         ex      de, hl
code:4DDF CD 4A 48   call    find_end_of_lowest_invader_row
code:4DDF CD E2 45   call    compare_video_addresses
code:4DE2 EB         ex      de, hl
code:4DE3 E5         push    hl
code:4DE4 F5         push    af
code:4DE5 FE FF      cp      #0xFF
code:4DE7 28 DF      jr      Z, delete_bomb_and_loop
code:4DE9 F1         pop     af
code:4DEA E1         pop     hl
code:4DEB 11 80 3F   ld      de, #video_ram+0x380      ; 2nd bottom row
code:4DEE CD E2 45   call    compare_video_addresses
code:4DF1 B7         or      a          ; possible shield hit?
code:4DF2 C2 27 4B   jp      NZ, handle_base_hit      ; no, must be player base
code:4DF5 CD 10 48   call    erode_shield_from_bomb
code:4DF8 E5         push    hl
code:4DF9 F5         push    af
code:4DFA 18 CC      jr      delete_bomb_and_loop
code:4DFA          ; End of function update_bombs
code:4DFA
code:4DFC
code:4DFC          ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000000000000000000
code:4DFC
code:4DFC
code:4DFC          zero_bullet_tbl:          ; CODE XREF: code:4EFB↑p
code:4DFC 21 EA 4E   ld      hl, #bomb_tbl
code:4DFF 11 EB 4E   ld      de, #bomb_tbl+1
code:4E02 01 0C 00   ld      bc, #0xC
code:4E05 36 00      ld      (hl), #0
code:4E07 ED B0      ldir
code:4E09 C9         ret
code:4E09          ; End of function zero_bullet_tbl
code:4E09
code:4E0A
code:4E0A          ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000000000000000000
code:4E0A
code:4E0A
code:4E0A          handle_bullet_destroys_bomb:          ; CODE XREF: update_bullet+DA↑p
code:4E0A E5         push    hl
code:4E0B D9         exx
code:4E0C D1         pop     de
code:4E0D DD 21 EA 4E ld      ix, #bomb_tbl
code:4E11 06 04      ld      b, #4          ; max bombs
code:4E13
code:4E13          loc_0_4E13:          ; CODE XREF:
handle_bullet_destroys_bomb+1A↑j
code:4E13 DD 6E 00   ld      l, 0(ix)
code:4E16 DD 66 01   ld      h, 1(ix)      ; bomb address
code:4E19 CD E2 45   call    compare_video_addresses
code:4E1C FE 20      cp      #0x20 ; ' '      ; hit?
code:4E1E CA 26 4E   jp      Z, loc_0_4E26      ; yes, skip
code:4E21 CD 2A 4E   call    add_3_to_ix      ; next bullet data
code:4E24 10 ED      djnz    loc_0_4E13      ; loop thru all bullets
code:4E26
code:4E26          loc_0_4E26:          ; CODE XREF:
handle_bullet_destroys_bomb+14↑j
code:4E26 D9         exx
code:4E27 C3 53 4D      jp      delete_bomb      ; returns
code:4E27          ; End of function handle_bullet_destroys_bomb
code:4E27
code:4E2A
code:4E2A          ; 0000000000000000 S U B R O U T I N E
0000000000000000000000000000000000000000000000000000000000000000
code:4E2A
code:4E2A
code:4E2A          add_3_to_ix:          ; CODE XREF:
handle_drop_new_bomb+49↑p
code:4E2A DD 23          ; update_bombs+D↑p ...
code:4E2A          inc      ix

```

[illegible]

```

code:4E72 DD 6E 01      ld      l, 1(ix)
code:4E75 DD 66 02      ld      h, 2(ix)          ; cursor position
code:4E78 22 20 40      ld      (0x4020), hl          ; set ROM variable
code:4E7B CD 2A 4E      call    add_3_to_ix
code:4E7E 18 E9         jr      loc_0_4E69          ; next character
code:4E80                ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4E80
code:4E80                loc_0_4E80:                ; CODE XREF:
print_slow_and_check_for_R_key+A↑j
code:4E80 CD 33 00      call    0x33          ; display character
code:4E83 01 00 05      ld      bc, #1280          ; ~20ms
code:4E86 CD 60 00      call    0x60          ; delay
code:4E89 CD D9 45      call    check_for_R_key
code:4E8C DD 23         inc     ix          ; next character
code:4E8E 18 D9         jr      loc_0_4E69          ; loop
code:4E8E                ; End of function print_slow_and_check_for_R_key
code:4E8E
code:4E90
code:4E90                ; 0000000000000000 S U B R O U T I N E
00000000000000000000000000000000000000000000000000000000000000000000
code:4E90
code:4E90
code:4E90                display_lives_left:                ; CODE XREF:
update_score_and_chk_bonus_life+24↑p
code:4E90 E5           ; code:49E9↑p ...
code:4E90                push     hl
code:4E91 D5           push     de
code:4E92 C5           push     bc
code:4E93 F5           push     af
code:4E94 2A 20 40      ld      hl, (0x4020)          ; current cursor position
code:4E97 E5           push     hl
code:4E98 21 D0 3F      ld      hl, #video_ram+0x3D0          ; cursor position
code:4E9B 22 20 40      ld      (0x4020), hl
code:4E9E 3A 0F 43      ld      a, (no_lives)
code:4EA1 3D           dec      a          ; any lives left?
code:4EA2 28 14         jr      Z, wipe_all_ship_icons
code:4EA4 47           ld      b, a          ; number of lives
code:4EA5
code:4EA5                loc_0_4EA5:                ; CODE XREF:
display_lives_left+1B↑j
code:4EA5 21 CF 4E      ld      hl, #base_icon
code:4EA8 CD E5 44      call    display_message
code:4EAB 10 F8         djnz   loc_0_4EA5          ; loop thru all icons
code:4EAD 3A 0F 43      ld      a, (no_lives)
code:4EB0 47           ld      b, a
code:4EB1 3E 04         ld      a, #4
code:4EB3 90           sub     b          ; no. icons to wipe
code:4EB4 28 0C         jr      Z, loc_0_4EC2          ; none, skip
code:4EB6 18 02         jr      wipe_ship_icons
code:4EB8                ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4EB8
code:4EB8                wipe_all_ship_icons:                ; CODE XREF:
display_lives_left+12↑j
code:4EB8 06 03         ld      b, #3          ; max 3 ship icons
code:4EBA
code:4EBA                wipe_ship_icons:                ; CODE XREF:
display_lives_left+26↑j
code:4EBA 21 CB 4E      ld      hl, #blank_x3          ; display_lives_left+30↑j
code:4EBA                ld      hl, #blank_x3
code:4EBD CD E5 44      call    display_message
code:4EC0 10 F8         djnz   wipe_ship_icons
code:4EC2
code:4EC2                loc_0_4EC2:                ; CODE XREF:
display_lives_left+24↑j
code:4EC2 E1           pop     hl
code:4EC3 22 20 40      ld      (0x4020), hl          ; restore cursor position
code:4EC6 F1           pop     af
code:4EC7 C1           pop     bc
code:4EC8 D1           pop     de
code:4EC9 E1           pop     hl
code:4ECA C9           ret
code:4ECA                ; End of function display_lives_left
code:4ECA
code:4ECA                ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

```

```

code:4ECB 20 20 20 00 blank_x3:.db 0x20, 0x20, 0x20, 0          ; DATA XREF:
display_lives_left+2A↑o
code:4ECF 88 8E 8C 00 base_icon:.db 0x88, 0x8E, 0x8C, 0        ; DATA XREF:
display_lives_left+15↑o
code:4ECF                                     ; handle_drop_new_bomb+5F↑o
code:4ED3 5C          bomb_characters:.db 0x5C ; \                ; bomb characters
code:4ED4 56          .db 0x56 ; V
code:4ED5 2A          .db 0x2A ; *
code:4ED6 47 4F 4F 44+aGoodLuck:.ascii 'GOOD LUCK'            ; DATA XREF:
display_GOOD_LUCK+D↑o
code:4ED6 20 4C 55 43+          .db 0
code:4EE0 20 20 20 20 20+blank_x9:.db 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0
code:4EE0 20 20 20 20+          ; DATA XREF:
display_GOOD_LUCK+1B↑o
code:4EEA D1 3E 56          bomb_tbl:.db 0xD1, 0x3E, 0x56      ; DATA XREF:
handle_drop_new_bomb+3D↑o
code:4EEA                                     ; update_bombs+1↑o ...
code:4EED DF 00 56          .db 0xDF, 0, 0x56
code:4EF0 92 3F 56          .db 0x92, 0x3F, 0x56
code:4EF3 00 00 00          .db 0, 0, 0
code:4EF6 00          .db 0 ;
code:4EF7 00          .db 0 ;
code:4EF8 00          .db 0 ;
code:4EF9 00          .db 0 ;
code:4EFA 00          .db 0 ;
code:4EFB                                     ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4EFB
code:4EFB          init_turn:                                     ; CODE XREF: code:4AAD↑j
code:4EFB CD FC 4D          call    zero_bullet_tbl
code:4EFE AF          xor      a
code:4EFF 32 12 43          ld      (unused_4312), a
code:4F02 32 20 43          ld      (game_timer), a
code:4F05
code:4F05          game_loop:                                     ; CODE XREF: code:4F95↑j
code:4F05 3A 40 38          ld      a, (0x3840)                    ; read keyboard
code:4F08 57          ld      d, a
code:4F09 01 00 02          ld      bc, #0x200                    ; ~7.5ms
code:4F0C CD 60 00          call    0x60                        ; delay
code:4F0F 3A 40 38          ld      a, (0x3840)                    ; read keyboard
code:4F12 AA          xor      d
code:4F13 E6 80          and     #0x80 ; 'e'                    ; space - changed state?
code:4F15 20 0F          jr      NZ, loc_0_4F26                    ; yes, skip
code:4F17 3A 22 43          ld      a, (keybd_state)                ; last read
code:4F1A 5F          ld      e, a
code:4F1B AA          xor      d                    ; changed state?
code:4F1C A2          and     d                    ; pressed?
code:4F1D E6 80          and     #0x80 ; 'e'                    ; space only
code:4F1F 7A          ld      a, d
code:4F20 32 22 43          ld      (keybd_state), a                ; store keyboard state
code:4F23 C4 E5 46          call    NZ, handle_fire                    ; yes, call
code:4F26
code:4F26          loc_0_4F26:                                     ; CODE XREF: code:4F15↑j
code:4F26 3A 20 43          ld      a, (game_timer)
code:4F29 E6 03          and     #3                    ; time to move player?
code:4F2B CC 88 48          call    Z, check_and_handle_move ; yes, call
code:4F2E 3A 11 43          ld      a, (bullet_active)
code:4F31 B7          or      a                    ; fired?
code:4F32 28 08          jr      NZ, loc_0_4F3C                    ; no, skip
code:4F34 3A 20 43          ld      a, (game_timer)
code:4F37 E6 03          and     #3                    ; time to move bullet?
code:4F39 CC 1C 47          call    Z, update_bullet                    ; yes, call
code:4F3C
code:4F3C          loc_0_4F3C:                                     ; CODE XREF: code:4F32↑j
code:4F3C 3A 20 43          ld      a, (game_timer)
code:4F3F E6 07          and     #7                    ; time to move UFO?
code:4F41 CC F6 48          call    Z, update_ufo                    ; yes, call
code:4F44 CD BF 48          call    check_and_start_ufo
code:4F47 3A 10 43          ld      a, (ufo_active)
code:4F4A B7          or      a                    ; on-screen?
code:4F4B 20 0D          jr      NZ, loc_0_4F5A                    ; yes, skip
code:4F4D 3A 0C 43          ld      a, (ufo_timer)
code:4F50 FE 80          cp      #0x80 ; 'e'                    ; time to wipe bonus?
code:4F52 20 06          jr      NZ, loc_0_4F5A                    ; no, skip
code:4F54 21 00 3C          ld      hl, #video_ram
code:4F57 CD 7E 4C          call    clear_video_line_HL                ; wipe bonus
code:4F5A

```



```

code:4F5A          loc_0_4F5A:                ; CODE XREF: code:4F4B↑j
code:4F5A 00                ; code:4F52↑j
code:4F5A          nop
code:4F5B 3A 20 43          ld      a, (game_timer)
code:4F5E E6 0F            and     #0xF                ; time to update bombs?
code:4F60 F5              push    af
code:4F61 CC 5F 4D          call    Z, update_bombs      ; yes, call
code:4F64 F1              pop     af
code:4F65 CC 8A 4C          call    Z, handle_drop_new_bomb
code:4F68 21 20 43          ld      hl, #game_timer
code:4F6B 34              inc     (hl)                ; increment game timer
code:4F6C 3A 21 43          ld      a, (invader_timer)
code:4F6F 3D              dec     a                ; tick
code:4F70 F5              push    af                ; time to move invaders?
code:4F71 CC 48 4B          call    Z, animate_and_move_invaders ; yes, call
code:4F74 F1              pop     af
code:4F75 F5              push    af
code:4F76 C4 60 4E          call    NZ, delay_1_5ms
code:4F79 F1              pop     af                ; invader timer expired?
code:4F7A 20 06            jr      NZ, loc_0_4F82      ; no, skip
code:4F7C 3A 01 43          ld      a, (invaders_left)
code:4F7F 87              add     a, a
code:4F80 D6 01            sub     #1                ; calc new invader timer
code:4F82          loc_0_4F82:                ; CODE XREF: code:4F7A↑j
code:4F82 32 21 43          ld      (invader_timer), a
code:4F85 3A 11 43          ld      a, (bullet_active)
code:4F88 B7              or      a                ; fired?
code:4F89 20 0A            jr      NZ, loc_0_4F95      ; yes, skip
code:4F8B 3A 00 43          ld      a, (fire_throttle)
code:4F8E B7              or      a
code:4F8F 28 04            jr      Z, loc_0_4F95
code:4F91 3D              dec     a
code:4F92 32 00 43          ld      (fire_throttle), a
code:4F95          loc_0_4F95:                ; CODE XREF: code:4F89↑j
code:4F95 C3 05 4F          jp      game_loop          ; code:4F8F↑j
code:4F95          ;
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
code:4F98 43 4F 50 59+aCopyrightC1979:..ascii 'COPYRIGHT (C) 1979, '
code:4FAC B0 37 2F 31+      .db 0xB0, 0x37, 0x2F, 0x31, 0x33, 0x42, 0x59, 0x20, 0x54, 0x52
code:4FAC 33 42 59 20+      .db 0x53, 0x2D, 0x42, 0x4D, 0x20, 0x4B, 0x4F, 0x47, 0x41, 0x4E
code:4FAC 54 52 53 2D+      .db 0x45, 0x49, 0, 0x81, 0x5F, 0x7A, 0xFE, 0x30, 0x28, 2, 0x77
code:4FAC 42 4D 20 4B+      .db 0x23, 0x7B, 0xE, 0xA, 0x10, 0xEC, 0xC6, 0x30, 0x77, 0x23,
0x36
code:4FAC 4F 47 41 4E+      .db 3, 0xE1, 6, 0x20, 0x3E, 0, 0x3D, 0x20, 4, 0x77, 0x23, 0x10
code:4FAC 45 49 00 81+      .db 0xFC, 0xAF, 0xC9, 0x3A, 0xC4, 0x4E, 0x6F, 0xCB, 0xA6, 0x5A
code:4FAC 5F 7A FE 30+      .db 0x23, 0x56, 0xCD, 0x82, 0x4E, 0x7B, 0xF, 0xF, 0xF, 0xE6,
0x1F
code:4FAC 28 02 77 23+      .db 0xC5, 0x21, 0xC0, 0x4D, 0x4F, 6, 0, 9, 0x7B
code:4FAC 7B 0E 0A 10+; end of 'code'
code:4FAC EC C6 30 77+
code:4FAC 23 36 03 E1+; end of file

```