


```

code:43BE 4E 44 59 20+      .db 0                                ; code:45BB|o
code:43BE 45 4C 45 43+      .db 9, 1Eh, 3Eh                      ; DATA XREF: code:45A0|o
code:43D4 3C 2D 2D 2D+a30Points:.text "<---- 30 POINTS"
code:43E6 09 1E 3E          .db 9, 1Eh, 3Eh                      ; DATA XREF: code:45A0|o
code:43E9 3C 2D 2D 2D+a20Points:.text "<---- 20 POINTS"
code:43FB 09 9E 3E          .db 9, 9Eh, 3Eh                      ; DATA XREF: code:45A0|o
code:43FE 3C 2D 2D 2D+a10Points:.text "<---- 10 POINTS"
code:4410 09 1E 3F          .db 9, 1Eh, 3Fh                      ; DATA XREF: code:45A0|o
code:4413 3C 2D 2D 2D+a_Mystery:.text "<---- ? MYSTERY"
code:4413 2D 20 20 20+      .db 0
code:4427 50 52 45 53+aPressZKeyToMov:.text "PRESS "            ; DATA XREF: code:45C7|o
code:4427 53 20 22 5A+      .db 22h
code:4427 22 20 4B 45+      .text "Z"
code:4427 59 20 54 4F+      .db 22h
code:4427 20 4D 4F 56+      .text " KEY TO MOVE LEFT"
code:4441 09 13 3D          .db 9, 13h, 3Dh                      ; DATA XREF: code:45C7|o
code:4444 50 52 45 53+aPressXKeyToMov:.text "PRESS "
code:4444 53 20 22 58+      .db 22h
code:4444 22 20 4B 45+      .text "X"
code:4444 59 20 54 4F+      .db 22h
code:4444 20 4D 4F 56+      .text " KEY TO MOVE RIGHT"
code:445F 09 93 3D          .db 9, 93h, 3Dh                      ; DATA XREF: code:45C7|o
code:4462 50 52 45 53+aPressKeyToFire:.text "PRESS "
code:4462 53 20 22 20+      .db 22h
code:4462 22 20 4B 45+      .text " "
code:4462 59 20 54 4F+      .db 22h
code:4462 20 46 49 52+      .text " KEY TO FIRE !"
code:4479 09 13 3E          .db 9, 13h, 3Eh                      ; DATA XREF: code:45C7|o
code:447C 50 52 45 53+aPressRKeyToSta:.text "PRESS "
code:447C 53 20 22 52+      .db 22h
code:447C 22 20 4B 45+      .text "R"
code:447C 59 20 54 4F+      .db 22h
code:447C 20 53 54 41+      .text " KEY TO START"
code:447C 52 54 00          .db 0
code:4493 47 20 41 20+aGAMEOVER:.text "G A M E - O V E R"        ; DATA XREF: code:494B|o
code:4493 4D 20 45 20+      .db 0
code:44A5 20 20 53 43+aScore00000High:.text " SCORE 00000 HIGH-SCORE"
code:44A5 4F 52 45 20+      .db 0                                ; DATA XREF: code:4544|o
code:44A5 20 30 30 30+      .text " 00000"
code:44A5 30 30 20 20+      .db 0
code:44E5 ; SUBROUTINE
code:44E5
code:44E5
code:44E5
code:44E5
code:44E5 C5
code:44E5 display_message: ; CODE XREF: code:4547|p
code:44E5 push bc ; code:455B|p ...
code:44E6
code:44E6 loc_0_44E6: ; CODE XREF: display_message+15|j
code:44E6 7E ld a, (hl) ; get character
code:44E7 B7 or a ; finished?
code:44E8 28 19 jr z, loc_0_4503 ; yes, exit
code:44EA FE 01 cp 1
code:44EC 20 0E jr nz, loc_0_44FC
code:44EE 06 05 ld b, 5 ; 5 characters to print
code:44F0 3E 08 ld a, 8 ; backspace
code:44F2
code:44F2 loc_0_44F2: ; CODE XREF: display_message+12|j
code:44F2 D5 push de
code:44F3 CD 33 00 call 33h ; display character
code:44F6 D1 pop de
code:44F7 10 F9 djnz loc_0_44F2 ; loop
code:44F9
code:44F9 loc_0_44F9: ; CODE XREF: display_message+1C|j
code:44F9 23 inc hl ; next character
code:44FA 18 EA jr loc_0_44E6 ; loop
code:44FC
code:44FC loc_0_44FC: ; CODE XREF: display_message+7|j
code:44FC D5 push de
code:44FD CD 33 00 call 33h ; display character
code:4500 D1 pop de
code:4501 18 F6 jr loc_0_44F9 ; loop
code:4503
code:4503 loc_0_4503: ; CODE XREF: display_message+3|j
code:4503 C1 pop bc
code:4504 C9 ret
code:4504 ; End of function display_message
code:4505
code:4505 ; SUBROUTINE
code:4505
code:4505
code:4505
code:4505 display_message_slowly: ; CODE XREF: display_message_slowly+11|j
code:4505 7E ld a, (hl) ; code:494E|p
code:4505 or a ; get character
code:4507 B7 ret z ; done?
code:4508 D5 push de ; yes, exit
code:4509 C5 push bc
code:450A CD 33 00 call 33h ; display character
code:450D 01 00 23 ld bc, 2300h ; ~130ms
code:4510 CD 60 00 call 60h ; 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:4518
code:4518 ; SUBROUTINE
code:4518
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, 3BFFh ; start of video (-1)
code:451C 06 40 ld b, 40h ; characters/line
code:451E
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, 0Fh ; 15 lines
code:4523 3E 20 ld a, 20h ; space

```

```

code:4525 11 40 00      ld      de, 40h ; '@'          ; next line address
code:4528
code:4528      loc_0_4528:      ; CODE XREF: wipe_screen_left_to_right_slow+12|j
code:4528      ld      (hl), a          ; display character
code:4529      add     hl, de          ; next line
code:452A      djnz   loc_0_4528      ; loop all lines
code:452C      cd      60 4E          call    delay_1_5ms
code:452F      ei          pop     hl
code:4530      ci          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      ; -----
code:4535
code:4535      START:
code:4535      F3          di
code:4536      3E 0F          ld      a, 0Fh
code:4538      CD 33 00          call   33h          ; display character
code:453B      31 8A 42          ld      sp, 428Ah
code:453E      21 C0 3F          ld      hl, video_ram+3C0h          ; cursor position
code:4541      22 20 40          ld      (4020h), 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, 20h ; ' '
code:454C      32 FF 3F          ld      (video_ram+3FFh), a
code:454F
code:454F      CD 18 45          attract_loop:      ; CODE XREF: code:45D6|j
code:454F      ; code:4960|j
code:4552      call    wipe_screen_left_to_right_slow
code:4552      ld      hl, video_ram+394h          ; cursor position
code:4555      22 20 40          ld      (4020h), 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+9Eh          ; cursor position
code:4561      22 20 40          ld      (4020h), 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+192h          ; cursor position
code:456D      22 20 40          ld      (4020h), hl
code:4570      21 23 43          ld      hl, invader_30pt
code:4573      CD E5 44          call   display_message
code:4576      21 12 3E          ld      hl, video_ram+212h          ; cursor position
code:4579      22 20 40          ld      (4020h), hl
code:457C      21 30 43          ld      hl, invader_20pt
code:457F      CD E5 44          call   display_message
code:4582      21 92 3E          ld      hl, video_ram+292h          ; cursor poition
code:4585      22 20 40          ld      (4020h), hl
code:4588      21 3D 43          ld      hl, invader_10pt
code:458B      CD E5 44          call   display_message
code:458E      21 12 3F          ld      hl, video_ram+312h          ; cursor position
code:4591      22 20 40          ld      (4020h), hl
code:4594      21 4A 43          ld      hl, ufo
code:4597      CD E5 44          call   display_message
code:459A      21 9E 3D          ld      hl, video_ram+19Eh          ; cursor position
code:459D      22 20 40          ld      (4020h), hl
code:45A0      21 D4 43          ld      hl, a30Points          ; "<---- 30 POINTS"
code:45A3      CD 66 4E          call   print_slow_and_check_for_R_key
code:45A6      01 FF FF          ld      bc, 0FFFFh          ; ~1s
code:45A9      C0 00 00          call   60h          ; delay
code:45AC      C0 00 00          call   60h          ; delay
code:45AF      CD 18 45          call   wipe_screen_left_to_right_slow
code:45B2      CD D9 45          call   check_for_R_key
code:45B5      21 94 3F          ld      hl, video_ram+394h
code:45B8      22 20 40          ld      (4020h), hl
code:45BB      21 BE 43          ld      hl, aTandyElectroni          ; ** TANDY ELECTRONICS **
code:45BE      CD E5 44          call   display_message
code:45C1      21 93 3C          ld      hl, video_ram+93h
code:45C4      22 20 40          ld      (4020h), hl
code:45C7      21 27 44          ld      hl, aPressZKeyToMov          ; "PRESS \"Z\" KEY TO MOVE LEFT"
code:45CA      CD 66 4E          call   print_slow_and_check_for_R_key
code:45CD      01 FF FF          ld      bc, 0FFFFh          ; ~1s
code:45D0      C0 00 00          call   60h          ; delay
code:45D3      C0 00 00          call   60h          ; delay
code:45D6      C3 4F 45          jp      attract_loop
code:45D9
code:45D9      ; SUBROUTINE
code:45D9
code:45D9      check_for_R_key:      ; CODE XREF: code:45B2|p
code:45D9      3A 04 38          ; print_slow_and_check_for_R_key+23|p
code:45D9      ld      a, (3804h)          ; read keyboard
code:45DC      FE 04          cp      4          ; "R"?
code:45DE      CA C2 49          jp      z, start_game          ; yes, skip
code:45E1      C9          ret
code:45E1      ; End of function check_for_R_key
code:45E1
code:45E2      ; SUBROUTINE
code:45E2
code:45E2      check_bullet_hit:      ; CODE XREF: update_bullet+26|p
code:45E2      E5          ; update_bullet+36|p ...
code:45E2      push     hl          ; bullet address
code:45E3      D5          push     de          ; video address to check
code:45E4      7C          ld      a, h
code:45E5      BA          cp      d          ; same MSB?
code:45E6      28 06          jr      z, loc_0_45EE          ; yes, continue
code:45E8      30 0A          jr      nc, loc_0_45F4          ; continue if bullet below
code:45EA
code:45EA      no_hit:      ; CODE XREF: check_bullet_hit+10|j
code:45EA      AF          xor      a          ; flag no hit
code:45EB
code:45EB      loc_0_45EB:      ; CODE XREF: check_bullet_hit+14|j
code:45EB      D1          ; check_bullet_hit+18|j
code:45EB      pop      de
code:45EC      E1          pop      hl
code:45ED      C9          ret
code:45EE
code:45EE      loc_0_45EE:      ; CODE XREF: check_bullet_hit+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: check_bullet_hit+6 j
code:45F4      ld      a, 0FFh      ; flag bullet right/below
code:45F4 3E FF      jr      loc_0_45EB
code:45F6 18 F3      ;
code:45F8
code:45F8
code:45F8      loc_0_45F8:      ; CODE XREF: check_bullet_hit+E j
code:45F8      ld      a, 20h ; ' '      ; flag bullet match
code:45FA 18 EF      jr      loc_0_45EB
code:45FA      ; End of function check_bullet_hit
code:45FA
code:45FC
code:45FC      ; SUBROUTINE
code:45FC
code:45FC      animate_invaders:      ; CODE XREF: sub_0_4B48+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, 3Fh ; '?'      ; characters/line-1
code:4603
code:4603      animate_top_row:      ; CODE XREF: animate_invaders+10 j
code:4603      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      0Fh      ; invert top 4 pixels
code:460A 77      ld      (hl), a      ; display
code:460B
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:4624
code:4625      ; SUBROUTINE
code:4625
code:4625      animate_invader_row:      ; CODE XREF: animate_invaders+16 p
code:4625      ; animate_invaders+1C p ...
code:4625      push     hl      ; invader row addr
code:4626 06 40      ld      b, 40h ; '@'      ; characters/line
code:4628
code:4628      loc_0_4628:      ; CODE XREF: animate_invader_row+15 j
code:4628      ld      a, (hl)      ; get character from video
code:4629 FE 8C      cp      8Ch ; 'i'
code:462B 28 09      jr      z, loc_0_4636
code:462D FE 83      cp      83h ; 'a'
code:462F 20 08      jr      nz, loc_0_4639      ; not part of animation, skip
code:4631 3E 8C      ld      a, 8Ch ; 'i'      ; 0x83->0x8C
code:4633 77      ld      (hl), a      ; display
code:4634 18 03      jr      loc_0_4639
code:4636
code:4636      loc_0_4636:      ; CODE XREF: animate_invader_row+6 j
code:4636      ld      a, 83h ; 'a'      ; 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      ; 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      ; SUBROUTINE
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, 3Eh ; '>'      ; 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, 20h ; ' '      ; 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      ; 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), 20h ; ' '      ; display space at left-hand byte
code:4658
code:4658      loc_0_4658:      ; CODE XREF: move_video_line_right_HL+14 j
code:4658      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), 20h ; ' '      ; display space
code:465F C9      ret
code:465F      ; End of function move_video_line_right_HL
code:465F
code:4660      ; SUBROUTINE
code:4660
code:4660

```

```

code:4660      move_video_line_left_HL:                ; CODE XREF: check_and_handle_move+2B|p
code:4660      ; update_ufo+32|p ...
code:4660 E5      push    hl
code:4661      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, 20h ; ' '                 ; space
code:466D 12      ld      (de), a                      ; display space LH byte
code:466E      loc_0_466E:                            ; CODE XREF: 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), 20h ; ' '              ; space RH byte
code:4676      loc_0_4676:                            ; CODE XREF: move_video_line_left_HL+10|j
code:4676      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, 3Fh ; '?'
code:467E 19      add     hl, de
code:467F 36 20    ld      (hl), 20h ; ' '              ; display space on end of row
code:4681 C9      ret
code:4681      ; End of function move_video_line_left_HL
code:4682
code:4682      ; SUBROUTINE
code:4682
code:4682      add_10_to_score:                        ; CODE XREF: 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+3CCh          ; 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: add_10_to_score+8|j
code:46A1 C1      ; add_10_to_score+E|j ...
code:46A1      pop     bc
code:46A2      pop     hl
code:46A3 C9      ret
code:46A3      ; End of function add_10_to_score
code:46A4
code:46A4      ; SUBROUTINE
code:46A4
code:46A4      add_1_to_score_digit:                    ; CODE XREF: add_10_to_score+5|p
code:46A4      ; add_10_to_score+B|p ...
code:46A4 7E      ld      a, (hl)                          ; get score digit
code:46A5 FE 39    cp      39h ; '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
code:46AC      flag_carry:                            ; CODE XREF: add_1_to_score_digit+3|j
code:46AC 36 30    ld      (hl), 30h ; '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:46B0
code:46B0      ; SUBROUTINE
code:46B0
code:46B0      zero_score:                            ; CODE XREF: add_10_to_score+1C|p
code:46B0 21 C9 3F  ld      hl, video_ram+3C9h          ; code:49EF|p
code:46B0      ld      b, 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), 30h ; '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:46BB
code:46BB      ; SUBROUTINE
code:46BB
code:46BB      update_score_and_chk_bonus_life:        ; CODE XREF: update_score_and_chk_bonus_life+27|j
code:46BB      ; update_bullet+50|p ...
code:46BB CD 82 46 call    add_10_to_score
code:46BB      ld      hl, video_ram+3CCh          ; tens digit of score
code:46BE 21 CC 3F  ld      a, (hl)                      ; get digit
code:46C1 7E      cp      30h ; '0'                    ; 0?
code:46C2 FE 30    jr      nz, loc_0_46E2                ; no, exit
code:46C4 20 1C    djnz   hl                            ; hundreds digit
code:46C6 2B      dec     hl                            ; get digit
code:46C7 7E      ld      a, (hl)                          ; 5?
code:46C8 FE 35    cp      35h ; '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      31h ; '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       30h ; '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      ret
code:46E4 C9      ; End of function update_score_and_chk_bonus_life
code:46E4
code:46E5
code:46E5      ; SUBROUTINE
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, (byte_0_4300)
code:46ED B7      or       a
code:46EE C0      ret       nz
code:46EF 3E 14    ld       a, 20
code:46F1 32 00 43 ld       (byte_0_4300), a
code:46F4 3A 10 43 ld       a, (ufo_active)
code:46F7 B7      or       a
code:46F8 20 08    jr       nz, loc_0_4702
code:46FA 3A 0D 43 ld       a, (ufo_dir)
code:46FD EE 01    xor      1
code:46FF 32 0D 43 ld       (ufo_dir), a
code:4702
code:4702      loc_0_4702:      ; CODE XREF: handle_fire+13[j
code:4702 3E 01    ld       a, 1
code:4704 32 11 43 ld       (bullet_active), a ; flag fired
code:4707 D9      exx
code:4708 2A 13 43 ld       hl, (base_centre)
code:470B 11 C0 FF ld       de, 0FFC0h ; -64
code:470E 19      add      hl, de ; video address of row above
code:470F 22 1E 43 ld       (bullet_addr), hl ; save
code:4712 7E      ld       a, (hl) ; get character from video
code:4713 FE 20    cp       20h ; ' ' ; space?
code:4715 C2 07 48 jr       nz, loc_0_4807 ; no, skip
code:4718 36 5B    ld       (hl), 5Bh ; '{' ; display player bullet
code:471A D9      exx
code:471B C9      ret
code:471B      ; End of function handle_fire
code:471B
code:471C
code:471C      ; SUBROUTINE
code:471C
code:471C      update_bullet:      ; CODE XREF: update_bullet+EC[j
code:471C D9      ; code:4F39[p
code:471C      exx
code:471D 2A 1E 43 ld       hl, (bullet_addr)
code:4720 7E      ld       a, (hl) ; get character from video
code:4721 FE 5B    cp       5Bh ; '[' ; player bullet?
code:4723 20 1A    jr       nz, handle_bullet_hit ; no, skip
code:4725 36 20    ld       (hl), 20h ; ' ' ; display space
code:4727 11 C0 FF ld       de, 0FFC0h ; -64
code:472A 19      add      hl, de ; address of row above
code:472B CB 54    bit      2, h ; off the top of the screen?
code:472D 28 61    jr       z, delete_bullet ; yes, skip
code:472F 7E      ld       a, (hl) ; get character from video
code:4730 FE 80    cp       80h ; 'Ç' ; graphic space?
code:4732 28 04    jr       z, display_bullet ; yes, skip
code:4734 FE 20    cp       20h ; ' ' ; space?
code:4736 20 07    jr       nz, handle_bullet_hit ; no, skip
code:4738
code:4738      display_bullet:      ; CODE XREF: update_bullet+16[j
code:4738 36 5B    ld       (hl), 5Bh ; '{' ; display player bullet
code:4738
code:473A      loc_0_473A:      ; CODE XREF: update_bullet+E8[j
code:473A 22 1E 43 ld       (bullet_addr), hl ; update bullet address
code:473D D9      exx
code:473E C9      ret
code:473F
code:473F      handle_bullet_hit:      ; CODE XREF: update_bullet+7[j
code:473F 11 40 3C    ; update_bullet+1A[j
code:473F      ld       de, video_ram+40h ; 2nd line on display
code:4742 CD E2 45 call     check_bullet_hit
code:4745 B7      or       a ; bullet in top line now?
code:4746 28 54    jr       z, check_and_handle_ufo_hit ; yes, skip
code:4748 CB 7E    bit      7, (hl) ; graphic character?
code:474A CA EA 47 jr       z, loc_0_47EA ; no, skip
code:474D E5      push     hl
code:474E CD 4A 48 call     sub_0_484A
code:4751 D1      pop      de
code:4752 CD E2 45 call     check_bullet_hit
code:4755 B7      or       a
code:4756 D5      push     de
code:4757 E1      pop      hl
code:4758 CA 0B 48 jr       z, loc_0_480B
code:475B CD 28 4D call     sub_0_4D28
code:475E 22 20 40 ld       (4020h), hl ; cursor position
code:4761 7E      ld       a, (hl) ; get MSB
code:4762 06 03    ld       b, 3 ; default to 30 pts
code:4764 FE A0    cp       0A0h ; 'Á'
code:4766 28 04    jr       z, loc_0_476C
code:4768 38 01    jr       c, loc_0_476B ; 20 pts if ???
code:476A 05      dec      b ; otherwise 10 pts
code:476B
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      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:481C A1          and      c
code:481D B9          cp       c
code:481E 20 02      jr        nz, loc_0_4822
code:4820 3E 20      ld        a, 20h ; ' '
code:4822
code:4822 FE 80      loc_0_4822:          ; CODE XREF: sub_0_4810+81j
code:4822          ; sub_0_4810+E1j
code:4822          cp       80h ; 'Ç'
code:4824 20 02      jr        nz, loc_0_4828
code:4826 3E 20      ld        a, 20h ; ' '
code:4828
code:4828          loc_0_4828:          ; CODE XREF: sub_0_4810+141j
code:4828          ld        (hl), a
code:4829 C1          pop      bc
code:482A E1          pop      hl
code:482B C3 57 4D      jp        loc_0_4D57
code:482B          ; End of function sub_0_4810
code:482E
code:482E          ; SUBROUTINE
code:482E
code:482E
code:482E          sub_0_482E:          ; CODE XREF: update_bullet+EF1p
code:482E          push     hl
code:482F C5          push     bc
code:4830 7E          ld        a, (hl)
code:4831 4F          ld        c, a
code:4832 3E 8F      ld        a, 8Fh ; 'À'
code:4834 A1          and      c
code:4835 B9          cp       c
code:4836 20 08      jr        nz, loc_0_4840
code:4838 3E 83      ld        a, 83h ; 'Ã'
code:483A A1          and      c
code:483B B9          cp       c
code:483C 20 02      jr        nz, loc_0_4840
code:483E 3E 20      ld        a, 20h ; ' '
code:4840
code:4840          loc_0_4840:          ; CODE XREF: sub_0_482E+81j
code:4840          ; sub_0_482E+E1j
code:4840          cp       80h ; 'Ç'
code:4842 20 02      jr        nz, loc_0_4846
code:4844 3E 20      ld        a, 20h ; ' '
code:4846
code:4846          loc_0_4846:          ; CODE XREF: sub_0_482E+141j
code:4846          ld        (hl), a
code:4847 C1          pop      bc
code:4848 E1          pop      hl
code:4849 C9          ret
code:4849          ; End of function sub_0_482E
code:484A
code:484A          ; SUBROUTINE
code:484A
code:484A
code:484A          sub_0_484A:          ; CODE XREF: update_bullet+321p
code:484A          ; sub_0_4D5F+7D1p
code:484A          ld        hl, (row_4_invader_addr)
code:484D 7C          ld        a, h
code:484E B7          or        a
code:484F 20 11      jr        nz, loc_0_4862
code:4851 2A 06 43      ld        hl, (row_3_invader_addr)
code:4854 7C          ld        a, h
code:4855 B7          or        a
code:4856 20 0A      jr        nz, loc_0_4862
code:4858 2A 04 43      ld        hl, (row_2_invader_addr)
code:485B 7C          ld        a, h
code:485C B7          or        a
code:485D 20 03      jr        nz, loc_0_4862
code:485F 2A 02 43      ld        hl, (row_1_invader_addr)
code:4862
code:4862          loc_0_4862:          ; CODE XREF: sub_0_484A+51j
code:4862          ; sub_0_484A+C1j ...
code:4862          push     de
code:4863 C5          push     bc
code:4864 11 3F 00      ld        de, 3Fh ; '?'
code:4867 19          add      hl, de
code:4868 06 3F      ld        b, 3Fh ; '?'
code:486A
code:486A          loc_0_486A:          ; CODE XREF: sub_0_484A+251j
code:486A          bit       7, (hl)
code:486C 20 03      jr        nz, loc_0_4871
code:486E 2B          dec      hl
code:486F 10 F9      djnz     loc_0_486A
code:4871
code:4871          loc_0_4871:          ; CODE XREF: sub_0_484A+221j
code:4871          pop      bc
code:4872 D1          pop      de
code:4873 C9          ret
code:4873          ; End of function sub_0_484A
code:4874
code:4874          ;
code:4874          loc_0_4874:          ; CODE XREF: update_bullet+661j
code:4874          jp        new_wave
code:4877
code:4877          ; SUBROUTINE
code:4877
code:4877
code:4877          sub_0_4877:          ; CODE XREF: sub_0_4B48+C1p
code:4877          ; code:4B8B1p
code:4877          ld        de, 40h ; '@'
code:487A 06 0D      ld        b, 0Dh
code:487C
code:487C          loc_0_487C:          ; CODE XREF: sub_0_4877+A1j
code:487C          bit       7, (hl)
code:487E 20 06      jr        nz, loc_0_4886
code:4880 19          add      hl, de
code:4881 10 F9      djnz     loc_0_487C
code:4883 3E 20      ld        a, 20h ; ' '
code:4885 C9          ret
code:4886
code:4886          ;
code:4886          loc_0_4886:          ; CODE XREF: sub_0_4877+71j
code:4886          xor      a
code:4887 C9          ret
code:4887          ; End of function sub_0_4877

```



```

code:4887
code:4888
code:4888
code:4888
code:4888
code:4888
code:4888 3A 08 38
code:488B E6 05
code:488D C8
code:488E FE 04
code:4890 30 16
code:4892 3A BB 3F
code:4895 CB 7F
code:4897 C0
code:4898 D9
code:4899 21 80 3F
code:489C CD 3E 46
code:489F 2A 13 43
code:48A2 23
code:48A3 22 13 43
code:48A6
code:48A6
code:48A6 D9
code:48A7 C9
code:48A8
code:48A8
code:48A8 C0
code:48A9 3A 84 3F
code:48AC CB 7F
code:48AE C0
code:48AF D9
code:48B0 21 80 3F
code:48B3 CD 60 46
code:48B6 2A 13 43
code:48B9 2B
code:48BA 22 13 43
code:48BD 18 E7
code:48BD
code:48BD
code:48BF
code:48BF
code:48BF
code:48BF
code:48BF
code:48BF 3A 0C 43
code:48C2 3C
code:48C3 32 0C 43
code:48C6 C0
code:48C7 3A 01 43
code:48CA FE 08
code:48CC D8
code:48CD 3A 10 43
code:48D0 B7
code:48D1 C0
code:48D2 D9
code:48D3 3E 41
code:48D5 32 0B 43
code:48D8 3A 0D 43
code:48DB B7
code:48DC 28 05
code:48DE 21 00 3C
code:48E1 18 03
code:48E3
code:48E3
code:48E3
code:48E3 21 3A 3C
code:48E6
code:48E6
code:48E6 22 20 40
code:48E9 21 4A 43
code:48EC CD E5 44
code:48EF 3E 01
code:48F1 32 10 43
code:48F4 D9
code:48F5 C9
code:48F5
code:48F5
code:48F6
code:48F6
code:48F6
code:48F6
code:48F6
code:48F6 3A 10 43
code:48F9 B7
code:48FA C8
code:48FB D9
code:48FC 21 00 3C
code:48FF 06 3F
code:4901
code:4901
code:4901 7E
code:4902 FE BB
code:4904 28 09
code:4906 FE B7
code:4908 28 09
code:490A
code:490A
code:490A 23
code:490A
code:490B 10 F4
code:490D 18 08
code:490F
code:490F
code:490F
code:490F 36 B7
code:4911 18 F7
code:4913
code:4913
code:4913
code:4913 36 BB
code:4915 18 F3
code:4917
code:4917
code:4917
code:4917 3A 0D 43

; SUBROUTINE

check_and_handle_move:
    ld a, (3808h)
    and 5
    ret z
    cp 4
    jr nc, loc_0_48A8
    ld a, (video_ram+3BBh)
    bit 7, a
    ret nz
    exx
    ld hl, video_ram+380h
    call move_video_line_right_HL
    ld hl, (base_centre)
    inc hl
    ld (base_centre), hl
loc_0_48A6:
    exx
    ret

;

loc_0_48A8:
    ret nz
    ld a, (video_ram+384h)
    bit 7, a
    ret nz
    exx
    ld hl, video_ram+380h
    call move_video_line_left_HL
    ld hl, (base_centre)
    dec hl
    ld (base_centre), hl
    jr loc_0_48A6
; End of function check_and_handle_move

; SUBROUTINE

check_and_start_ufo:
    ld a, (byte_0_430C)
    inc a
    ld (byte_0_430C), a
    ret nz
    ld a, (invaders_left)
    cp 8
    ret c
    ld a, (ufo_active)
    or a
    ret nz
    ld a, 65
    ld (ufo_TTL), a
    ld a, (ufo_dir)
    or a
    jr z, loc_0_48E3
    ld hl, video_ram
    jr loc_0_48E6
loc_0_48E3:
    ld hl, video_ram+3Ah
loc_0_48E6:
    ld (4020h), hl
    ld hl, ufo
    call display_message
    ld a, 1
    ld (ufo_active), a
    exx
    ret
; End of function check_and_start_ufo

; SUBROUTINE

update_ufo:
    ld a, (ufo_active)
    or a
    ret z
    exx
    ld hl, video_ram
    ld b, 63
loc_0_4901:
    ld a, (hl)
    cp 0BBh ; 'q'
    jr z, loc_0_490F
    cp 0B7h ; 'A'
    jr z, loc_0_4913
loc_0_490A:
    inc hl
    djnz loc_0_4901
    jr loc_0_4917
loc_0_490F:
    ld (hl), 0B7h ; 'A'
    jr loc_0_490A
loc_0_4913:
    ld (hl), 0BBh ; 'q'
    jr loc_0_490A
loc_0_4917:
    ld a, (ufo_dir)

```

```

code:491A B7          or      a          ; left?
code:491B 28 08      jr      z, loc_0_4925      ; yes, skip
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      loc_0_492B
code:4925
code:4925
code:4925 loc_0_4925:          ; 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 loc_0_492B:          ; CODE XREF: update_ufo+2D↑j
code:492B 3A 0B 43    ld      a, (ufo_TTL)
code:492E 3D          dec      a          ; ufo still active?
code:492F 32 0B 43    ld      (ufo_TTL), a
code:4932 28 02      jr      z, loc_0_4936      ; no, skip
code:4934
code:4934 loc_0_4934:          ; CODE XREF: update_ufo+44↑j
code:4934 D9          exx
code:4935 C9          ret
code:4936
code:4936
code:4936 loc_0_4936:          ; CODE XREF: update_ufo+3C↑j
code:4936 AF          xor      a          ; flag ufo inactive
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
code:493C
code:493C game_over:          ; CODE XREF: code:4AB4↑j
code:493C 31 8A 42          ; code:4B45↑j
code:493C          ld      sp, 428Ah
code:493F 21 19 3C    ld      hl, video_ram+19h      ; cursor position
code:4942 22 20 40    ld      (4020h), hl
code:4945 21 00 3C    ld      hl, video_ram      ; start of video
code:4948 CD 7E 4C    call    clear_video_line_HL
code:494B 21 93 44    ld      hl, aGAMEOVER      ; "G A M E - O V E R"
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, 0FFFFh      ; ~1s
code:4957 CD 60 00    call    60h      ; delay
code:495A CD 60 00    call    60h      ; delay
code:495D CD 60 00    call    60h      ; delay
code:4960 C3 4F 45    jp      attract_loop
code:4963
code:4963 ; SUBROUTINE
code:4963
code:4963
code:4963 display_object_x10:          ; 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      ; 10 objects to display
code:4968 D5          push    de
code:4969 11 40 00      ld      de, 40h ; '@'
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_object_x10+1C↑j
code:4970 22 20 40      ld      (4020h), hl      ; cursor position
code:4973 D5          push    de
code:4974 E5          push    hl
code:4975 EB          ex      de, hl
code:4976 CD E5 44    call    display_message
code:4979 E1          pop     hl
code:497A 11 06 00      ld      de, 6
code:497D 19          add     hl, de
code:497E D1          pop     de
code:497F 10 EF      djnz   loc_0_4970
code:4981 C1          pop     bc
code:4982 D1          pop     de
code:4983 E1          pop     hl
code:4984 C9          ret
code:4984 ; End of function display_object_x10
code:4984
code:4985
code:4985 ; SUBROUTINE
code:4985
code:4985
code:4985 sub_0_4985:          ; CODE XREF: update_bullet+6F↑p
code:4985 E5          push    hl
code:4986 D5          push    de
code:4987 C5          push    bc
code:4988 2A 02 43      ld      hl, (row_1_invader_addr)
code:498B CD B0 49    call    sub_0_49B0
code:498E 22 02 43      ld      (row_1_invader_addr), hl
code:4991 2A 04 43      ld      hl, (row_2_invader_addr)
code:4994 CD B0 49    call    sub_0_49B0
code:4997 22 04 43      ld      (row_2_invader_addr), hl
code:499A 2A 06 43      ld      hl, (row_3_invader_addr)
code:499D CD B0 49    call    sub_0_49B0
code:49A0 22 06 43      ld      (row_3_invader_addr), hl
code:49A3 2A 08 43      ld      hl, (row_4_invader_addr)
code:49A6 CD B0 49    call    sub_0_49B0
code:49A9 22 08 43      ld      (row_4_invader_addr), hl
code:49AC C1          pop     bc
code:49AD D1          pop     de
code:49AE E1          pop     hl
code:49AF C9          ret
code:49AF ; End of function sub_0_4985
code:49AF
code:49B0
code:49B0
code:49B0
code:49B0 sub_0_49B0:          ; CODE XREF: sub_0_4985+6↑p
code:49B0 E5          ; sub_0_4985+F↑p ...
code:49B0          push    hl
code:49B1 06 3F      ld      b, 3Fh ; '?'
code:49B3
code:49B3 loc_0_49B3:          ; CODE XREF: sub_0_49B0+9↑j
code:49B3 7E          ld      a, (hl)
code:49B4 CB 7F      bit     7, a

```

```

code:49B6 23          inc     hl
code:49B7 20 06        jr      nz, loc_0_49BF
code:49B9 10 F8        djnz   loc_0_49B3
code:49BB E1        pop     hl
code:49BC 26 00        ld      h, 0
code:49BE C9        ret
code:49BF
code:49BF
code:49BF loc_0_49BF:          ; CODE XREF: sub_0_49B0+71j
code:49BF E1        pop     hl
code:49C0 19        add     hl, de
code:49C1 C9        ret
code:49C1 ; End of function sub_0_49B0
code:49C2
code:49C2
code:49C2 start_game:          ; CODE XREF: check_for_R_key+51j
code:49C2 AF        xor      a
code:49C3 32 0C 43    ld      (byte_0_430C), a
code:49C6 32 15 43    ld      (byte_0_4315), a
code:49C9 32 12 43    ld      (byte_0_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      (byte_0_4322), a
code:49DB 3C        inc     a
code:49DC 32 0A 43    ld      (byte_0_430A), a
code:49DF 3E 05        ld      a, 5
code:49E1 32 0C 43    ld      (byte_0_430C), 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, 428Ah
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:48741j
code:49F5 31 8A 42    ld      sp, 428Ah
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:4A021j
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, 40h ; '@' ; 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:4A161j
code:4A1A 04          inc     b
code:4A1B 19        add     hl, de
code:4A1C
code:4A1C loc_0_4A1C:          ; CODE XREF: code:4A141j
code:4A1C EB        ex      de, hl
code:4A1D
code:4A1D calc_invader_row_addr: ; CODE XREF: code:4A401j
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:4A3D1j
code:4A28 DD 6E 00        ld      l, (ix+0)
code:4A2B DD 66 01        ld      h, (ix+1)
code:4A2E 19        add     hl, de ; calc video address for invader row
code:4A2F FD 75 00        ld      (iy+0), l
code:4A32 FD 74 01        ld      (iy+1), 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+309h ; cursor position
code:4A48 22 20 40        ld      (4020h), 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+317h ; cursor position
code:4A55 22 20 40        ld      (4020h), hl
code:4A58 E1        pop     hl ; shield #2
code:4A59 E5        push   hl
code:4A5A CD E5 44        call   display_message
code:4A5D 21 24 3F        ld      hl, video_ram+324h ; cursor position
code:4A60 22 20 40        ld      (4020h), hl
code:4A63 E1        pop     hl ; shield #3
code:4A64 E5        push   hl
code:4A65 CD E5 44        call   display_message
code:4A68 21 31 3F        ld      hl, video_ram+331h ; cursor position
code:4A6B 22 20 40        ld      (4020h), hl ; shield #4
code:4A6E E1        pop     hl
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_object_x10 ; 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_object_x10 ; 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_object_x10 ; display 3rd row of invaders
code:4A8D 2A 08 43        ld      hl, (row_4_invader_addr)

```

```

code:4A90 CD 63 49      call    display_object_x10      ; 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      (byte_0_4321), a
code:4A9B
code:4A9B      loc_0_4A9B:
code:4A9B 21 84 3F      ld      hl, video_ram+384h      ; CODE XREF: code:4AD9|j
code:4A9E 22 20 40      ld      (4020h), hl            ; cursor position
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+386h
code:4AAA 22 13 43      ld      (base_centre), hl      ; center of base
code:4AAD C3 FB 4E      jp      loc_0_4EFB
code:4AB0
code:4AB0      loc_0_4AB0:
code:4AB0 3A 0F 43      ld      a, (no_lives)          ; CODE XREF: code:4B2A|j
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      (byte_0_4312), a
code:4AC4 CD 3B 4D      call    restore_space_characters
code:4AC7 21 80 3F      ld      hl, video_ram+380h
code:4ACA CD 7E 4C      call    clear_video_line_HL
code:4ACD 01 FF FF      ld      bc, 0FFFFh
code:4AD0 CD 60 00      call    60h
code:4AD3 CD 60 00      call    60h
code:4AD6 31 8A 42      ld      sp, 428Ah
code:4AD9 C3 9B 4A      jp      loc_0_4A9B
code:4ADC
code:4ADC      ; ██████████ S U B R O U T I N E ██████████
code:4ADC
code:4ADC      invert_display:
code:4ADC 21 00 3C      ld      hl, video_ram          ; CODE XREF: sub_0_4B1A+4|p
code:4ADF 01 00 04      ld      bc, 400h              ; sub_0_4B1A+B|j ...
code:4AE2
code:4AE2      loc_0_4AE2:
code:4AE2 7E          ld      a, (hl)                ; CODE XREF: invert_display+1C|j
code:4AE3 FE 20      cp      20h ; ' '             ; get character
code:4AE5 20 02      jr      nz, loc_0_4AE9        ; space?
code:4AE7 36 80      ld      (hl), 80h ; 'Ç'       ; no, skip
code:4AE9
code:4AE9      loc_0_4AE9:
code:4AE9 CB 7E      bit     7, (hl)                ; CODE XREF: invert_display+9|j
code:4AEB 28 07      jr      z, loc_0_4AF4        ; graphics character?
code:4AED 7E      ld      a, (hl)                ; no, skip
code:4AEE 2F      cpl     ; get character
code:4AEF CB FF      set     7, a                   ; invert
code:4AF1 CB B7      res     6, a                   ; make graphics character
code:4AF3 77      ld      (hl), a                ; 1st block of graphics characters
code:4AF4
code:4AF4      loc_0_4AF4:
code:4AF4 23      inc     hl                     ; CODE XREF: invert_display+F|j
code:4AF5 0B      dec     bc                     ; next video address
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
code:4AFA      ; End of function invert_display
code:4AFB
code:4AFB      ; ██████████ S U B R O U T I N E ██████████
code:4AFB
code:4AFB      sub_0_4AFB:
code:4AFB 2A 13 43      ld      hl, (base_centre)      ; CODE XREF: sub_0_4B1A|p
code:4AFE 2B      dec     hl
code:4AFF 2B      dec     hl
code:4B00 C9      ret
code:4B00
code:4B00      ; End of function sub_0_4AFB
code:4B01
code:4B01      ; ██████████ S U B R O U T I N E ██████████
code:4B01
code:4B01      sub_0_4B01:
code:4B01 3E A6      ld      a, 0A6h ; 'A'         ; CODE XREF: sub_0_4B1A+8|p
code:4B03 06 00      ld      b, 0
code:4B05
code:4B05      loc_0_4B05:
code:4B05 C5      push    bc                     ; CODE XREF: sub_0_4B01+16|j
code:4B06 E5      push    hl
code:4B07 06 05      ld      b, 5
code:4B09 EE 3F      xor     3Fh ; '?'
code:4B0B
code:4B0B      loc_0_4B0B:
code:4B0B 77      ld      (hl), a                ; CODE XREF: sub_0_4B01+C|j
code:4B0E 23      inc     hl
code:4B0D 10 FC      djnz    loc_0_4B0B
code:4B0F
code:4B0F      loc_0_4B0F:
code:4B0F E3      ex      (sp), hl               ; CODE XREF: sub_0_4B01+12|j
code:4B10 E3      ex      (sp), hl
code:4B11 E3      ex      (sp), hl
code:4B12 E3      ex      (sp), hl
code:4B13 10 FA      djnz    loc_0_4B0F
code:4B15 E1      pop     hl
code:4B16 C1      pop     bc
code:4B17 10 EC      djnz    loc_0_4B05
code:4B19 C9      ret
code:4B19
code:4B19      ; End of function sub_0_4B01
code:4B1A
code:4B1A      ; ██████████ S U B R O U T I N E ██████████
code:4B1A
code:4B1A      sub_0_4B1A:
code:4B1A CD FB 4A      call    sub_0_4AFB             ; CODE XREF: code:4B27|p
code:4B1D E5      push    hl
code:4B1E CD DC 4A      call    invert_display
code:4B21 E1      pop     hl

```

```

code:4B22 CD 01 4B      call    sub_0_4B01
code:4B25 18 B5         jr      invert_display
code:4B25             ; End of function sub_0_4B1A
code:4B25             ;
code:4B27
code:4B27
code:4B27 loc_0_4B27:          ; CODE XREF: sub_0_4D5F+93|j
code:4B27 CD 1A 4B      call    sub_0_4B1A
code:4B2A C3 B0 4A      jp      loc_0_4AB0
code:4B2D             ;
code:4B2D
code:4B2D loc_0_4B2D:          ; CODE XREF: code:4C02|j
code:4B2D 06 0A         ld      b, 0Ah
code:4B2F
code:4B2F loc_0_4B2F:          ; 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, 2710h
code:4B36 CD 60 00      call    60h
code:4B39 CD DC 4A      call    invert_display
code:4B3C 01 10 27      ld      bc, 2710h
code:4B3F CD 60 00      call    60h
code:4B42 C1           pop     bc
code:4B43 10 EA         djnz   loc_0_4B2F
code:4B45 C3 3C 49      jp      game_over
code:4B48
code:4B48             ; SUBROUTINE
code:4B48
code:4B48
code:4B48 sub_0_4B48:          ; 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, (byte_0_430A)
code:4B4E B7           or      a
code:4B4F 28 37         jr      z, loc_0_4B88
code:4B51 21 7F 3C      ld      hl, video_ram+7Fh
code:4B54 CD 77 48      call    sub_0_4877
code:4B57 B7           or      a
code:4B58 CA BE 4B      jp      z, loc_0_4BBE
code:4B5B 2A 08 43      ld      hl, (row_4_invader_addr)
code:4B5E CD 7A 4B      call    sub_0_4B7A
code:4B61 2A 06 43      ld      hl, (row_3_invader_addr)
code:4B64 CD 7A 4B      call    sub_0_4B7A
code:4B67 2A 04 43      ld      hl, (row_2_invader_addr)
code:4B6A CD 7A 4B      call    sub_0_4B7A
code:4B6D 2A 02 43      ld      hl, (row_1_invader_addr)
code:4B70 CD 7A 4B      call    sub_0_4B7A
code:4B73 CD FC 45      call    animate_invaders
code:4B76
code:4B76 loc_0_4B76:          ; CODE XREF: code:4BAC|j
code:4B76 C1           ; code:4C1D|j ...
code:4B76         pop     bc
code:4B77 D1           pop     de
code:4B78 E1           pop     hl
code:4B79 C9           ret
code:4B79             ; End of function sub_0_4B48
code:4B79
code:4B7A             ; SUBROUTINE
code:4B7A
code:4B7A
code:4B7A
code:4B7A sub_0_4B7A:          ; CODE XREF: sub_0_4B48+16|p
code:4B7A 7C           ; sub_0_4B48+1C|p ...
code:4B7A         ld      a, h
code:4B7B B7           or      a
code:4B7C C8           ret     z
code:4B7D CD 3E 46      call    move_video_line_right_HL
code:4B80 11 40 00      ld      de, 40h ; '@'
code:4B83 ED 52         sbc     hl, de
code:4B85 C3 3E 46      jp      move_video_line_right_HL
code:4B85             ; End of function sub_0_4B7A
code:4B85
code:4B85
code:4B85
code:4B85 loc_0_4B88:          ; CODE XREF: sub_0_4B48+7|j
code:4B85 21 40 3C      ld      hl, video_ram+40h
code:4B8B CD 77 48      call    sub_0_4877
code:4B8E B7           or      a
code:4B8F 28 38         jr      z, loc_0_4BC9
code:4B91 CD FC 45      call    animate_invaders
code:4B94
code:4B94 loc_0_4B94:          ; CODE XREF: code:4BC6|j
code:4B94 2A 08 43      ld      hl, (row_4_invader_addr)
code:4B97 CD AE 4B      call    sub_0_4BAE
code:4B9A 2A 06 43      ld      hl, (row_3_invader_addr)
code:4B9D CD AE 4B      call    sub_0_4BAE
code:4BA0 2A 04 43      ld      hl, (row_2_invader_addr)
code:4BA3 CD AE 4B      call    sub_0_4BAE
code:4BA6 2A 02 43      ld      hl, (row_1_invader_addr)
code:4BA9 CD AE 4B      call    sub_0_4BAE
code:4BAC 18 C8         jr      loc_0_4B76
code:4BAE
code:4BAE             ; SUBROUTINE
code:4BAE
code:4BAE
code:4BAE
code:4BAE sub_0_4BAE:          ; CODE XREF: code:4B97|p
code:4BAE 7C           ; code:4B9D|p ...
code:4BAE         ld      a, h
code:4BAF B7           or      a
code:4BB0 C8           ret     z
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, 40h ; '@'
code:4BB9 ED 52         sbc     hl, de
code:4BBB C3 60 46      jp      move_video_line_left_HL
code:4BBB             ; End of function sub_0_4BAE
code:4BBB
code:4BBB
code:4BBE
code:4BBE loc_0_4BBE:          ; CODE XREF: sub_0_4B48+10|j
code:4BBE 3A 0A 43      ld      a, (byte_0_430A)
code:4BC1 EE 01         xor     l
code:4BC3 32 0A 43      ld      (byte_0_430A), a
code:4BC6 C3 94 4B      jp      loc_0_4B94

```

```

code:4BC9      ; -----
code:4BC9
code:4BC9      loc_0_4BC9:      ; CODE XREF: code:4B8F|j
code:4BC9 3A 11 43      ld      a, (bullet_active)
code:4BCC B7          or      a
code:4BCD 28 05      jr      z, loc_0_4BD4      ; fired?
code:4BCF 2A 1E 43      ld      hl, (bullet_addr)      ; no, skip
code:4BD2 36 20      ld      (hl), 20h ; ' '      ; 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, (ix+0)
code:4BDD DD 66 01      ld      h, (ix+1)      ; hl = invader addr
code:4BE0 7C          ld      a, h
code:4BE1 B7          or      a
code:4BE2 C4 2F 4C      call    nz, sub_0_4C2E      ; any invaders left on this line?
code:4BE5 DD 2B      dec     ix      ; yes, call
code:4BE7 DD 2B      dec     ix
code:4BE9 10 EF      djnz    loc_0_4BDA      ; next invader row address
code:4BEB DD 21 08 43      ld      ix, row_4_invader_addr      ; loop thru 4 rows of invaders
code:4BEF 06 04      ld      b, 4
code:4BF1 11 80 3F      ld      de, video_ram+380h      ; 4 rows of invaders
code:4BF4
code:4BF4      loc_0_4BF4:      ; CODE XREF: code:4C0F|j
code:4BF4 DD 6E 00      ld      l, (ix+0)
code:4BF7 DD 66 01      ld      h, (ix+1)      ; HL = invader row addr
code:4BFA CD 55 4C      call    sub_0_4C55
code:4BFD CD E2 45      call    check_bullet_hit
code:4C00 FE 20      cp      20h ; ' '
code:4C02 CA 2D 4B      jp      z, loc_0_4B2D
code:4C05 DD 75 00      ld      (ix+0), l
code:4C08 DD 74 01      ld      (ix+1), h
code:4C0B DD 2B      dec     ix
code:4C0D DD 2B      dec     ix
code:4C0F 10 E3      djnz    loc_0_4BF4      ; loop thru 4 rows of invaders
code:4C11 3A 0A 43      ld      a, (byte_0_430A)
code:4C14 EE 01      xor     l
code:4C16 32 0A 43      ld      (byte_0_430A), a
code:4C19 3A 11 43      ld      a, (bullet_active)
code:4C1C B7          or      a
code:4C1D CA 76 4B      jp      z, loc_0_4B76      ; fired?
code:4C20 2A 1E 43      ld      hl, (bullet_addr)      ; no, skip
code:4C23 7E          ld      a, (hl)
code:4C24 FE 20      cp      20h ; ' '      ; get character from video
code:4C26 C2 76 4B      jp      nz, loc_0_4B76      ; space?
code:4C29 36 5B      ld      (hl), 5Bh ; '['      ; display player bullet
code:4C2B C3 76 4B      jp      loc_0_4B76
code:4C2E
code:4C2E      ; SUBROUTINE
code:4C2E
code:4C2E
code:4C2E      sub_0_4C2E:      ; CODE XREF: code:4BE2|p
code:4C2E C5          push    bc
code:4C2F E5          push    hl
code:4C30 11 3F 00      ld      de, 3Fh ; '?'
code:4C33 19          add     hl, de
code:4C34 E5          push    hl
code:4C35 13          inc     de
code:4C36 19          add     hl, de
code:4C37 EB          ex      de, hl
code:4C38 E1          pop     hl
code:4C39 06 80      ld      b, 80h ; 'Ç'
code:4C3B
code:4C3B      loc_0_4C3B:      ; CODE XREF: sub_0_4C2E+1A|j
code:4C3B 1A          ld      a, (de)
code:4C3C FE 80      cp      80h ; 'Ç'
code:4C3E 7E          ld      a, (hl)
code:4C3F 30 04      jr      nc, loc_0_4C45
code:4C41 FE 80      cp      80h ; 'Ç'
code:4C43 38 01      jr      c, loc_0_4C46
code:4C45
code:4C45      loc_0_4C45:      ; CODE XREF: sub_0_4C2E+11|j
code:4C45 12          ld      (de), a
code:4C46
code:4C46      loc_0_4C46:      ; CODE XREF: sub_0_4C2E+15|j
code:4C46 1B          dec     de
code:4C47 2B          dec     hl
code:4C48 10 F1      djnz    loc_0_4C3B
code:4C4A E1          pop     hl
code:4C4B 11 40 00      ld      de, 40h ; '@'
code:4C4E ED 52      sbc     hl, de
code:4C50 CD 7E 4C      call    clear_video_line_HL
code:4C53 C1          pop     bc
code:4C54 C9          ret
code:4C54      ; End of function sub_0_4C2E
code:4C55
code:4C55      ; SUBROUTINE
code:4C55
code:4C55      sub_0_4C55:      ; CODE XREF: code:4BFA|p
code:4C55 7C          ld      a, h
code:4C56 B7          or      a
code:4C57 C8          ret     z
code:4C58 D5          push    de
code:4C59 11 40 00      ld      de, 40h ; '@'
code:4C5C 19          add     hl, de
code:4C5D D1          pop     de
code:4C5E C9          ret
code:4C5E      ; End of function sub_0_4C55
code:4C5F
code:4C5F      ; SUBROUTINE
code:4C5F
code:4C5F      check_for_new_high_score:      ; CODE XREF: code:4951|p
code:4C5F 21 C9 3F      ld      hl, video_ram+3C9h      ; score
code:4C62 11 FA 3F      ld      de, video_ram+3FAh      ; 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
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+3C9h    ; source = score
code:4C75 11 FA 3F     ld      de, video_ram+3FAh    ; 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          ; SUBROUTINE
code:4C7E
code:4C7E          clear_video_line_HL:          ; CODE XREF: update_bullet+94|p
code:4C7E C5          ; code:4948|p ...
code:4C7E          push     bc
code:4C7F D5          push     de
code:4C80 06 40       ld      b, 40h ; '@'          ; characters/line
code:4C82
code:4C82          loc_0_4C82:          ; CODE XREF: clear_video_line_HL+7|j
code:4C82 36 20       ld      (hl), 20h ; ' '          ; 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          ; SUBROUTINE
code:4C8A
code:4C8A          sub_0_4C8A:          ; CODE XREF: code:4F65|p
code:4C8A D9          exx
code:4C8B 06 04       ld      b, 4
code:4C8D DD 21 08 43 ld      ix, row_4_invader_addr
code:4C91
code:4C91          loc_0_4C91:          ; CODE XREF: sub_0_4C8A+11|j
code:4C91 DD 7E 01     ld      a, (ix+1)
code:4C94 B7          or      a
code:4C95 20 08       jr      nz, loc_0_4C9F
code:4C97 DD 2B       dec     ix
code:4C99 DD 2B       dec     ix
code:4C9B 10 F4       djnz    loc_0_4C91
code:4C9D
code:4C9D          loc_0_4C9D:          ; CODE XREF: sub_0_4C8A+3B|j
code:4C9D D9          exx          ; sub_0_4C8A+4E|j ...
code:4C9D C9          ret
code:4C9F
code:4C9F          loc_0_4C9F:          ; CODE XREF: sub_0_4C8A+B|j
code:4C9F 2A 13 43       ld      hl, (base_centre)
code:4CA2 11 80 C0     ld      de, 0C080h
code:4CA5 19          add     hl, de
code:4CA6 EB          ex      de, hl
code:4CA7 DD 6E 00     ld      l, (ix+0)
code:4CAA DD 66 01     ld      h, (ix+1)
code:4CAD E5          push    hl
code:4CAE 21 03 00     ld      hl, 3
code:4CB1 CD 1D 4D     call    rand
code:4CB4 7D          ld      a, l
code:4CB5 E1          pop     hl
code:4CB6 FE 01       cp      1
code:4CB8 20 4B       jr      nz, loc_0_4D05
code:4CBA
code:4CBA          loc_0_4CBA:          ; CODE XREF: sub_0_4C8A+84|j
code:4CBA 19          add     hl, de
code:4CBB 11 80 FF     ld      de, 0FF80h
code:4CBE
code:4CBE          loc_0_4CBE:          ; CODE XREF: sub_0_4C8A+39|j
code:4CBE CB 7E          bit     7, (hl)
code:4CC0 20 05       jr      nz, loc_0_4CC7
code:4CC2 19          add     hl, de
code:4CC3 10 F9       djnz    loc_0_4CBE
code:4CC5 18 D6       jr      loc_0_4C9D
code:4CC7
code:4CC7          loc_0_4CC7:          ; CODE XREF: sub_0_4C8A+36|j
code:4CC7 DD 21 EA 4E     ld      ix, 4EEAh
code:4CCB 06 04       ld      b, 4
code:4CCD
code:4CCD          loc_0_4CCD:          ; CODE XREF: sub_0_4C8A+4C|j
code:4CCD DD 7E 01     ld      a, (ix+1)
code:4CD0 B7          or      a
code:4CD1 28 08       jr      z, loc_0_4CDB
code:4CD3 CD 2A 4E     call    add_3_to_ix
code:4CD6 10 F5       djnz    loc_0_4CCD
code:4CD8 C3 9D 4C     jp      loc_0_4C9D
code:4CDB
code:4CDB          loc_0_4CDB:          ; CODE XREF: sub_0_4C8A+47|j
code:4CDB CD 28 4D     call    sub_0_4D28
code:4CDE 11 82 00     ld      de, 82h ; 'é'
code:4CE1 19          add     hl, de
code:4CE2 E5          push    hl
code:4CE3 21 03 00     ld      hl, 3
code:4CE6 CD 1D 4D     call    rand
code:4CE9 11 D2 4E     ld      de, 4ED2h
code:4CEC 19          add     hl, de
code:4CED D1          pop     de
code:4CEE CB 7B          bit     7, e
code:4CF0 28 1E       jr      z, loc_0_4D10
code:4CF2
code:4CF2          loc_0_4CF2:          ; CODE XREF: sub_0_4C8A+8A|j
code:4CF2 1A          ld      a, (de)
code:4CF3 FE 20       cp      20h ; ' '
code:4CF5 20 A6       jr      nz, loc_0_4C9D

```



```

code:4CF7 7E          ld      a, (hl)
code:4CF8 12          ld      (de), a
code:4CF9 DD 73 00    ld      (ix+0), e
code:4CFC DD 72 01    ld      (ix+1), d
code:4CFF DD 77 02    ld      (ix+2), a
code:4D02 C3 9D 4C    jp      loc_0_4C9D
code:4D05
code:4D05
code:4D05          loc_0_4D05:          ; CODE XREF: sub_0_4C8A+2E↑j
code:4D05 E5          push     hl
code:4D06 21 40 00    ld      hl, 40h ; '@'
code:4D09 CD 1D 4D    call    rand
code:4D0C EB          ex       de, hl
code:4D0D E1          pop      hl
code:4D0E 18 AA          jr      loc_0_4CBA
code:4D10
code:4D10
code:4D10          loc_0_4D10:          ; CODE XREF: sub_0_4C8A+66↑j
code:4D10 EB          ex       de, hl
code:4D11 CB 7E          bit     7, (hl)
code:4D13 EB          ex       de, hl
code:4D14 28 DC          jr      z, loc_0_4CF2
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      sub_0_4810
code:4D1A          ; End of function sub_0_4C8A
code:4D1D
code:4D1D          ; ██████████ SUBROUTINE ██████████
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    14CCh          ; ROM RAND() function
code:4D22 CD 7F 0A    call    0A7Fh          ; 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:4D28
code:4D28          ; ██████████ SUBROUTINE ██████████
code:4D28
code:4D28
code:4D28          sub_0_4D28:          ; CODE XREF: update_bullet+3F↑p
code:4D28 D5          ; sub_0_4C8A+51↑p
code:4D28          push     de
code:4D29 7E          ld      a, (hl)
code:4D2A E6 30          and     30h ; '0'
code:4D2C 11 C0 FF    ld      de, 0FFC0h
code:4D2F 20 01          jr      nz, loc_0_4D32
code:4D31 19          add     hl, de
code:4D32
code:4D32          loc_0_4D32:          ; CODE XREF: sub_0_4D28+7↑j
code:4D32 CB 7E          ; sub_0_4D28+D↑j
code:4D32          bit     7, (hl)
code:4D34 2B          dec     hl
code:4D35 20 FB          jr      nz, loc_0_4D32
code:4D37 23          inc     hl
code:4D38 23          inc     hl
code:4D39 D1          pop      de
code:4D3A C9          ret
code:4D3A          ; End of function sub_0_4D28
code:4D3A
code:4D3A          ; ██████████ SUBROUTINE ██████████
code:4D3A
code:4D3A
code:4D3A          restore_space_characters:      ; CODE XREF: code:4AC4↑p
code:4D3B          ld      hl, video_ram
code:4D3E 01 C0 03    ld      bc, 3C0h          ; 15 lines (all but last)
code:4D41
code:4D41          loc_0_4D41:          ; CODE XREF: restore_space_characters+16↑j
code:4D41 CB 7E          ; graphics character?
code:4D43 28 05          jr      z, loc_0_4D4A    ; no, skip
code:4D45 7E          ld      a, (hl)          ; get character
code:4D46 FE 80          cp      80h ; 'Ç'        ; graphic space character?
code:4D48 20 02          jr      nz, loc_0_4D4C    ; no, skip
code:4D4A
code:4D4A          loc_0_4D4A:          ; CODE XREF: restore_space_characters+8↑j
code:4D4A          ld      (hl), 20h ; ' '    ; convert to space character
code:4D4C
code:4D4C          loc_0_4D4C:          ; CODE XREF: restore_space_characters+D↑j
code:4D4C          inc     hl              ; next video address
code:4D4D 0B          dec     bc
code:4D4E 78          ld      a, b
code:4D4F B1          or      c                ; done?
code:4D50 C8          ret      z                ; yes, return
code:4D51 18 EE          jr      loc_0_4D41        ; loop through 15 lines
code:4D51          ; End of function restore_space_characters
code:4D53
code:4D53          ; ██████████ SUBROUTINE ██████████
code:4D53
code:4D53
code:4D53          delete_bomb:          ; CODE XREF: sub_0_4D5F+20↑p
code:4D53 AF          ; sub_0_4D5F+60↑p ...
code:4D53          xor      a                ; zero bomb address
code:4D54 DD 77 01    ld      (ix+1), a
code:4D57
code:4D57          loc_0_4D57:          ; CODE XREF: sub_0_4810+1B↑j
code:4D57          ld      a, (byte_0_4312)
code:4D5A 3D          dec     a
code:4D5B 32 12 43    ld      (byte_0_4312), a
code:4D5E C9          ret
code:4D5E          ; End of function delete_bomb
code:4D5E
code:4D5E          ; ██████████ SUBROUTINE ██████████
code:4D5E
code:4D5E
code:4D5E          sub_0_4D5F:          ; CODE XREF: code:4F61↑p

```

```

code:4D5F D9      exx
code:4D60 DD 21 EA 4E      ld      ix, 4EEAh
code:4D64 06 04      ld      b, 4
code:4D66
code:4D66      loc_0_4D66:                                ; CODE XREF: sub_0_4D5F+10↑j
code:4D66 DD 7E 01      ld      a, (ix+1)
code:4D69 B7      or      a
code:4D6A 20 07      jr      nz, loc_0_4D73
code:4D6C
code:4D6C      loc_0_4D6C:                                ; CODE XREF: sub_0_4D5F+23↑j
code:4D6C CD 2A 4E      ; sub_0_4D5F+4B↑j ...
code:4D6C      call    add_3_to_ix
code:4D6F 10 F5      djnz   loc_0_4D66
code:4D71 D9      exx
code:4D72 C9      ret
code:4D73
code:4D73
code:4D73      loc_0_4D73:                                ; CODE XREF: sub_0_4D5F+B↑j
code:4D73 DD 6E 00      ld      l, (ix+0)
code:4D76 DD 66 01      ld      h, (ix+1)
code:4D79 DD 7E 02      ld      a, (ix+2)
code:4D7C BE      cp      (hl)
code:4D7D 28 05      jr      z, loc_0_4D84
code:4D7F CD 53 4D      call   delete_bomb
code:4D82 18 E8      jr      loc_0_4D6C
code:4D84
code:4D84
code:4D84      loc_0_4D84:                                ; CODE XREF: sub_0_4D5F+1E↑j
code:4D84 36 20      ld      (hl), 20h ; ' '
code:4D86 11 40 00      ld      de, 40h ; '@'
code:4D89 19      add     hl, de
code:4D8A DD 75 00      ld      (ix+0), l
code:4D8D DD 74 01      ld      (ix+1), h
code:4D90 E5      push    hl
code:4D91 11 C0 3F      ld      de, video_ram+3C0h
code:4D94 CD E2 45      call   check_bullet_hit
code:4D97 B7      or      a
code:4D98 F5      push    af
code:4D99 20 2D      jr      nz, loc_0_4DC8
code:4D9B F1      pop     af
code:4D9C E1      pop     hl
code:4D9D 7E      ld      a, (hl)
code:4D9E FE 5B      cp      5Bh ; '['
code:4DA0 28 0A      jr      z, loc_0_4DAC
code:4DA2 FE 81      cp      81h ; 'û'
code:4DA4 30 35      jr      nc, loc_0_4DDB
code:4DA6 DD 7E 02      ld      a, (ix+2)
code:4DA9 77      ld      (hl), a
code:4DAA 18 C0      jr      loc_0_4D6C
code:4DAC
code:4DAC
code:4DAC      loc_0_4DAC:                                ; CODE XREF: sub_0_4D5F+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, loc_0_4DC8
code:4DB9 20 14      jr      nz, loc_0_4DCF
code:4DBB AF      xor     a
code:4DBC 32 11 43      ld      (bullet_active), a
code:4DBF CD 53 4D      call   delete_bomb
code:4DC2 F1      pop     af
code:4DC3 E1      pop     hl
code:4DC4 36 20      ld      (hl), 20h ; ' '
code:4DC6 18 A4      jr      loc_0_4D6C
code:4DC8
code:4DC8
code:4DC8      loc_0_4DC8:                                ; CODE XREF: sub_0_4D5F+3A↑j
code:4DC8 CD 53 4D      ; sub_0_4D5F+58↑j ...
code:4DC8      call   delete_bomb
code:4DCB F1      pop     af
code:4DCC E1      pop     hl
code:4DCD 18 9D      jr      loc_0_4D6C
code:4DCF
code:4DCF
code:4DCF      loc_0_4DCF:                                ; CODE XREF: sub_0_4D5F+5A↑j
code:4DCF AF      xor     a
code:4DD0 32 11 43      ld      (bullet_active), a
code:4DD3 F1      pop     af
code:4DD4 E1      pop     hl
code:4DD5 DD 7E 02      ld      a, (ix+2)
code:4DD8 77      ld      (hl), a
code:4DD9 18 91      jr      loc_0_4D6C
code:4DDB
code:4DDB
code:4DDB      loc_0_4DDB:                                ; CODE XREF: sub_0_4D5F+45↑j
code:4DDB EB      ex      de, hl
code:4DDC CD 4A 48      call   sub_0_484A
code:4DDF CD E2 45      call   check_bullet_hit
code:4DE2 EB      ex      de, hl
code:4DE3 E5      push    hl
code:4DE4 F5      push    af
code:4DE5 FE FF      cp      0FFh
code:4DE7 28 DF      jr      z, loc_0_4DC8
code:4DE9 F1      pop     af
code:4DEA E1      pop     hl
code:4DEB 11 80 3F      ld      de, video_ram+380h
code:4DEE CD E2 45      call   check_bullet_hit
code:4DF1 B7      or      a
code:4DF2 C2 27 4B      jp      nz, loc_0_4B27
code:4DF5 CD 10 48      call   sub_0_4810
code:4DF8 E5      push    hl
code:4DF9 F5      push    af
code:4DFA 18 CC      jr      loc_0_4DC8
code:4DFA      ; End of function sub_0_4D5F
code:4DFC
code:4DFC
code:4DFC      ; ██████████ S U B R O U T I N E ██████████
code:4DFC
code:4DFC      zero_scratchpad_ram:                        ; CODE XREF: code:4EFB↑p
code:4DFC 21 EA 4E      ld      hl, 4EEAh
code:4DFF 11 EB 4E      ld      de, 4EEBh
code:4E02 01 0C 00      ld      bc, 0Ch

```



```

code:4E86 CD 60 00      call    60h                ; delay
code:4E89 DD D9 45      call    check_for_R_key
code:4E8C DD 23        inc     ix                ; next character
code:4E8E 18 D9        jr      loc_0_4E69        ; loop
; End of function print_slow_and_check_for_R_key

; SUBROUTINE

display_lives_left:    ; CODE XREF: update_score_and_chk_bonus_life+24|p
                      ; code:49E9|p ...
code:4E90 E5          push    hl
code:4E90          push    de
code:4E91 D5          push    bc
code:4E92 C5          push    af
code:4E93 F5          ld      hl, (4020h)        ; current cursor position
code:4E94 2A 20 40    ld      hl, video_ram+3D0h    ; cursor position
code:4E97 E5          ld      (4020h), hl
code:4E98 21 D0 3F    ld      a, (no_lives)
code:4E9B 2A 20 40    ld      a, (no_lives)
code:4E9E 3A 0F 43    dec     a                ; any lives left?
code:4EA1 3D          jr      z, wipe_all_ship_icons
code:4EA2 28 14      jr      b, a                ; number of lives
code:4EA4 47          ; CODE XREF: display_lives_left+1B|j
code:4EA5          ld      hl, mini_player
code:4EA5 21 CF 4E    call    display_message
code:4EA8 CD E5 44    djnz    loc_0_4EA5
code:4EAB 10 F8      ld      a, (no_lives)
code:4EAD 3A 0F 43    ld      b, a
code:4EB0 47          ld      a, 4
code:4EB1 3E 04      sub     b                ; no. icons to wipe
code:4EB3 90          jr      z, loc_0_4EC2        ; none, skip
code:4EB4 28 0C      jr      wiper_ship_icons
code:4EB6 18 02      ;
code:4EB8          ; CODE XREF: display_lives_left+12|j
code:4EB8 06 03      ld      b, 3                ; max 3 ship icons
code:4EBA          ; CODE XREF: display_lives_left+26|j
code:4EBA 21 CB 4E    ; display_lives_left+30|j
code:4EBA          ld      hl, blank_x3
code:4EBD CD E5 44    call    display_message
code:4EC0 10 F8      djnz    wiper_ship_icons
code:4EC2          ; CODE XREF: display_lives_left+24|j
code:4EC2          pop     hl
code:4EC3 22 20 40    ld      (4020h), 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
; End of function display_lives_left

;
code:4ECB 20 20 20 00 blank_x3:.db 20h, 20h, 20h, 0        ; DATA XREF: display_lives_left+2A|o
code:4ECF 88 8E 8C 00 mini_player:.db 88h, 8Eh, 8Ch, 0      ; DATA XREF: display_lives_left+15|o
code:4ED3 5C          .db 5Ch ; \
code:4ED4 56          .db 56h ; V
code:4ED5 2A          .db 2Ah ; *
code:4ED6 47 4F 4F 44+aGoodLuck:.text "GOOD LUCK"          ; DATA XREF: display_GOOD_LUCK+D|o
code:4ED6 20 4C 55 43+.db 0
code:4EE0 20 20 20 20+blank_x9:.db 20h, 20h, 20h, 20h, 20h, 20h, 20h, 20h, 0
code:4EE0 20 20 20 20+.db 0DFh, 0, 56h                ; DATA XREF: display_GOOD_LUCK+1B|o
code:4EEA D1 3E 56    bomb_tbl:.db 0D1h, 3Eh, 56h          ; DATA XREF: check_and_handle_bullet_hit_bomb+3|o
code:4EEB DF 00 56    .db 0DFh, 0, 56h
code:4EF0 92 3F 56    .db 92h, 3Fh, 56h
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          ; CODE XREF: code:4AAD|j
code:4EFB          call    zero_scratchpad_ram
code:4EFB CD FC 4D    xor     a
code:4EFE AF          ld      (byte_0_4312), a
code:4EFF 32 12 43    ld      (byte_0_4320), a
code:4F02 32 20 43    ; CODE XREF: code:4F95|j
code:4F05          ld      a, (3840h)        ; read keyboard
code:4F05 3A 40 38    ld      d, a
code:4F08 57          ld      bc, 200h        ; ~7.5ms
code:4F09 01 00 02    call    60h                ; delay
code:4F0C CD 60 00    ld      a, (3840h)        ; read keyboard
code:4F0F 3A 40 38    xor     d
code:4F12 AA          and     80h ; 'C'        ; space - changed state?
code:4F13 E6 80      jr      nz, loc_0_4F26        ; yes, skip
code:4F15 20 0F      ld      a, (byte_0_4322)    ; last saved keyboard state
code:4F17 3A 22 43    ld      d, a
code:4F1A 5F          xor     d                ; changed state?
code:4F1B AA          and     d                ; pressed?
code:4F1C A2          and     80h ; 'C'        ; space only
code:4F1D E6 80      ld      a, d
code:4F1F 7A          ld      (byte_0_4322), a    ; store keyboard state
code:4F20 32 22 43    call    nz, handle_fire    ; yes, call
code:4F23 C4 E5 46    ; CODE XREF: code:4F15|j
code:4F26          ld      a, (byte_0_4320)
code:4F26          and     3
code:4F29 E6 03      call    z, check_and_handle_move
code:4F2B CC 88 48    ld      a, (bullet_active)
code:4F2E 3A 11 43    or     a                ; fired?
code:4F31 B7          jr      z, loc_0_4F3C        ; no, skip
code:4F32 28 08      ld      a, (byte_0_4320)
code:4F34 3A 20 43    and     3
code:4F37 E6 03      call    z, update_bullet
code:4F39 CC 1C 47    ; CODE XREF: code:4F32|j
code:4F3C          ld      a, (byte_0_4320)
code:4F3C          and     7
code:4F41 CC F6 48    call    z, update_ufo

```

```

code:4F44 CD BF 48      call    check_and_start_ufo
code:4F47 3A 10 43      ld      a, (ufo_active)
code:4F4A B7           or      a
code:4F4B 20 0D      jr      nz, loc_0_4F5A
code:4F4D 3A 0C 43      ld      a, (byte_0_430C)
code:4F50 FE 80      cp      80h ; 'Ç'
code:4F52 20 06      jr      nz, loc_0_4F5A
code:4F54 21 00 3C      ld      hl, video_ram
code:4F57 CD 7E 4C      call    clear_video_line_HL
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, (byte_0_4320)
code:4F5E E6 0F      and     0Fh
code:4F60 F5           push    af
code:4F61 CC 5F 4D      call    z, sub_0_4D5F
code:4F64 F1           pop     af
code:4F65 CC 8A 4C      call    z, sub_0_4C8A
code:4F68 21 20 43      ld      hl, 4320h
code:4F6B 34           inc     (hl)
code:4F6C 3A 21 43      ld      a, (byte_0_4321)
code:4F6F 3D           dec     a
code:4F70 F5           push    af
code:4F71 CC 48 4B      call    z, sub_0_4B48
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
code:4F7A 20 06      jr      nz, loc_0_4F82
code:4F7C 3A 01 43      ld      a, (invaders_left)
code:4F7F 87           add     a, a
code:4F80 D6 01      sub     1
code:4F82
code:4F82                loc_0_4F82:                ; CODE XREF: code:4F7A↑j
code:4F82 32 21 43      ld      (byte_0_4321), a
code:4F85 3A 11 43      ld      a, (bullet_active)
code:4F88 B7           or      a
code:4F89 20 0A      jr      nz, loc_0_4F95      ; fired?
code:4F8B 3A 00 43      ld      a, (byte_0_4300)      ; yes, skip
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      (byte_0_4300), a
code:4F95
code:4F95                loc_0_4F95:                ; CODE XREF: code:4F89↑j
code:4F95 C3 05 4F      ; code:4F8F↑j
code:4F95                jp      game_loop
code:4F95
code:4F98 43 4F 50 59+aCopyrightC1979: .text "COPYRIGHT (C) 1979, "
code:4FAC B0 37 2F 31+      .db 0B0h, 37h, 2Fh, 31h, 33h, 42h, 59h, 20h, 54h, 52h, 53h, 2Dh
code:4FAC 33 42 59 20+      .db 42h, 4Dh, 20h, 4Bh, 4Fh, 47h, 41h, 4Eh, 45h, 49h, 0, 81h, 5Fh
code:4FAC 54 52 53 2D+      .db 7Ah, 0FEh, 30h, 28h, 2, 77h, 23h, 7Bh, 0Eh, 0Ah, 10h, 0ECh
code:4FAC 42 4D 20 4B+      .db 0C6h, 30h, 77h, 23h, 36h, 3, 0E1h, 6, 20h, 3Eh, 0, 3Dh, 20h
code:4FAC 4F 47 41 4E+      .db 4, 77h, 23h, 10h, 0FCh, 0AFh, 0C9h, 3Ah, 0C4h, 4Eh, 6Fh, 0CBh
code:4FAC 45 49 00 81+      .db 0A6h, 5Ah, 23h, 56h, 0CDh, 82h, 4Eh, 7Bh, 0Fh, 0Fh, 0Fh, 0E6h
code:4FAC 5F 7A FE 30+      .db 1Fh, 0C5h, 21h, 0C0h, 4Dh, 4Fh, 6, 0, 9, 7Bh
code:4FAC 28 02 77 23+; end of 'code'
code:4FAC 7B 0E 0A 10+
code:4FAC EC C6 30 77+
code:4FAC 23 36 03 E1+      .end

```