```
<?xml version='1.0'?>
<CEPEL_GEDWEB>
    <CEPEL_Metadados>
        <CEPEL_Documento>
            <CEPEL_DocumentoRevisaoId>9999999</CEPEL_DocumentoRevisaoId>
            <CEPEL_DocumentoId>9999999</CEPEL_DocumentoId>
            <CEPEL_DocumentoNome>GALAX -- FUNDAMENTOS DE COMPUTADORES
DIGITAIS</CEPEL_DocumentoNome>
            <CEPEL_DocumentoNomeArquivo>202502072040-GEDWEB-
GALAX_MAC.xml</CEPEL_DocumentoNomeArquivo>
            <CEPEL_DocumentoDescricao>Jogo Desenvolvido 100% em Assembler Z80 para a
Disciplina de Fundamentos de Computadores Digitais do Professor Orlando
(UERJ)</CEPEL_DocumentoDescricao>
            <CEPEL_DocumentoData>20250207 20:40</CEPEL_DocumentoData>
            <CEPEL_DocumentoRevisaoData>20250207 20:58</CEPEL_DocumentoRevisaoData>
        </CEPEL_Documento>
        <CEPEL_Autor>
            <CEPEL_AutorId>9999999</CEPEL_AutorId>
            <CEPEL_AutorNome>Luiz Marcio Faria de Aquino Viana, Pós-
D.Sc.</CEPEL_AutorNome>
            <CEPEL_AutorFormacao>Engenheiro Eletricista com Ênfase em Engenharia de
Sistemas e Computação</CEPEL_AutorNome>
            <CEPEL_AutorEmail>luiz.marcio.viana@gmail.com</CEPEL_AutorEmail>
            <CEPEL_AutorTelefone>(21)99983-7207</CEPEL_AutorTelefone>
        </CEPEL_Autor>
        <CEPEL_UsuarioDownload>
            <CEPEL_UsuarioId>9999999</CEPEL_UsuarioId>
            <CEPEL_UsuarioNome>Luiz Marcio Faria de Aquino Viana, Pós-
D.Sc.</CEPEL_UsuarioNome>
            <CEPEL_UsuarioEmail>luiz.marcio.viana@gmail.com</CEPEL_UsuarioEmail>
            <CEPEL_UsuarioTelefone>(21)99983-7207</CEPEL_UsuarioTelefone>
        </CEPEL_UsuarioDownload>
    </CEPEL_Metadados>
    <CEPEL_Data><![CDATA[
TITLE  "GALAX -- FUNDAMENTOS DE COMPUTADORES DIGITAIS"

        ASEG
        .Z80

;
; --- declaracao das constantes
;

BDOS        EQU 0005h       ; Endereco de execucao das funcoes
                    ;   BDOS no CP/M

CPM     EQU 00h     ; Funcao de inicializacao do CP/M
CONIO       EQU 06h     ; Funcao de entrada e saida direta
                    ;   no console
CRTOUT      EQU 09h     ; Funcao de impressao de string na
                    ;   tela do terminal

TIMEC       EQU 00800h      ; Constante de tempo do jogo

; codigo ASCII de caracteres

NULL        EQU 00h     ; Codigo do caracter NULL
LEFTKEY         EQU 2Ch     ; Codigo da seta p/esquerda
RIGHTKEY    EQU 2Eh     ; Codigo da seta p/direita
```

```
BS      EQU 20h    ; Codigo da tecla backspace

; crash status

CRON       EQU 00h
CROFF      EQU 0FFh

; fire/bomb status

FIREON     EQU 00h
FIREOFF        EQU 0FFh

; status das naves inimigas

DEAD       EQU 0FFh       ; Eliminada
CONVOY     EQU 00h    ; Comboio
ATTACK     EQU 01h    ; Atacando

; direcao de movimento do comboio

LEFT       EQU 00h    ; Movendo para esquerda
RIGHT      EQU 01h    ; Movendo para direita

; imagem grafica a ser apresentada

IMAGE0     EQU 00h    ; Apresenta a imagem inicial
IMAGE1     EQU 01h    ; Apresenta a outra imagem

;
; --- rotina de inicializacao
;

       ORG 0100h

Inicio:       LD  SP, stack + 0FFh
       LD  C, CRTOUT
       LD  DE, clear
       CALL   BDOS
       CALL   IniStar
Loop1:     CALL   MovStar
       LD  C, CRTOUT
       LD  DE, logo
       CALL   BDOS
       LD  DE, msgAnyKey
       CALL   BDOS
       CALL   Delay
       CALL   GetKey
       CP  NULL
       JR  Z, Loop1
       LD  C, CRTOUT
       LD  DE, clear
       CALL   BDOS
       LD  DE, base
       CALL   BDOS
       CALL   IniLife
       CALL   IniScore
       CALL   MovStar
       CALL   IniSilon
       CALL   IniGorak
```

```
       CALL   IniShip
       LD  A, 03h
       LD  (shiplife), A
       CALL   IniFire
       CALL   IniBomb
       LD  A, CROFF
       LD  (shipCrash), A
       CALL   IniExpl
       CALL   IniScore
Loop2:    CALL   MovStar
       CALL   MovSilon
       CALL   MovGorak
       CALL   MovShip
       CALL   MovBomb
       CALL   TstCrash
       CALL   MovFire
       CALL   TstFire
       CALL   MovExpl
       CALL   Delay
       CALL   MovBomb
       CALL   TstCrash
       CALL   MovFire
       CALL   TstFire
       CALL   MovExpl
       CALL   TstFase
       LD  A, (cnvD2)
       LD  (cnvD1), A
       LD  A, (cnvImg)
       CP  IMAGE0
       JR  NZ, Loop3
       LD  A, IMAGE1
       JR  Loop4
Loop3:    LD  A, IMAGE0
Loop4:    LD  (cnvImg), A
       LD  A, (shipCrash)
       CP  CRON
       JR  NZ, Loop2
       CALL   MovLife
       LD  C, CPM
       CALL   BDOS

; ------------------------------------------------------------------------------
;                                  S H I P
; ------------------------------------------------------------------------------

; --- rotina de inicializacao das variaveis de controle da nave

IniShip:  LD  A, 39h
       LD  (xOFship), A
       LD  B, 22h
       LD  C, A
       LD  DE, ship
       CALL   WriteXY
       RET

; --- rotina de movimento da nave de combate

MovShip:  LD  IX, xOFship
       CALL   GetKey
```

```
        CP  LEFTKEY
        CALL    Z, MovLeft
        CP  RIGHTKEY
        CALL    Z, MovRight
        CP  BS
        CALL    Z, DispFire
        CALL    PutShip
        RET

MovLeft:    PUSH    AF
        LD  A, (IX)
        CP  00h
        RET Z
        PUSH    AF
        CALL    ClrShip
        POP AF
        SUB 01h
        DAA
        LD  (IX), A
        POP AF
        RET

MovRight:   PUSH    AF
        LD  A, (IX)
        CP  78h
        RET Z
        PUSH    AF
        CALL    ClrShip
        POP AF
        ADD A, 01h
        DAA
        LD  (IX), A
        POP AF
        RET

DispFire:   LD  A, (fireStat)
        CP  FIREON
        RET Z
        LD  A, FIREON
        LD  (fireStat), A
        LD  A, (IX)
        ADD A, 01h
        DAA
        LD  (xOFfire), A
        LD  A, 22h
        LD  (yOFfire), A
        RET

ClrShip:    PUSH    IX
        LD  B, 22h
        LD  C, (IX)
        LD  DE, blank
        CALL    WriteXY
        POP IX
        RET

PutShip:    PUSH    IX
        LD  B, 22h
        LD  C, (IX)
```

```
       LD  DE, ship
       CALL   WriteXY
       POP IX
       RET

; ------------------------------------------------------------------------------
;                                   F I R E
; ------------------------------------------------------------------------------

; --- rotina de inicializacao do tiro do mocinho

IniFire:  LD  A, FIREOFF
       LD  (fireStat), A
       RET

; --- rotina de movimentacao do tiro do mocinho

MovFire:  LD  A, (fireStat)
       CP  FIREOFF
       RET Z
       LD  IX, xOFfire
       LD  IY, yOFfire
       CALL   ClrFire
       LD  A, (IY)
       CP  00h
       JR  NZ, MFire1
       LD  A, FIREOFF
       LD  (fireStat), A
       RET
MFire1:       SUB 01h
       DAA
       LD  (IY), A
       CALL   PutFire
       RET

ClrFire:  PUSH   IX
       PUSH   IY
       LD  B, (IY)
       LD  C, (IX)
       LD  DE, blk
       CALL   WriteXY
       POP IY
       POP IX
       RET

PutFire:  PUSH   IX
       PUSH   IY
       LD  B, (IY)
       LD  C, (IX)
       LD  DE, fire
       CALL   WriteXY
       POP IY
       POP IX
       RET

; ------------------------------------------------------------------------------
;                             E S T R E L A S
; ------------------------------------------------------------------------------
```

```
;
; --- rotina de inicializacao das variaveis de movimento das estrelas
;

IniStar:  LD  IX, starX
       LD  IY, xOFstar
       LD  B, 44
IStar1:       LD  A, (IX)
       LD  (IY), A
       INC IX
       INC IY
       DJNZ   IStar1
       LD  IX, starY
       LD  IY, yOFstar
       LD  B, 44
IStar2:       LD  A, (IX)
       LD  (IY), A
       INC IX
       INC IY
       DJNZ   IStar2
       RET

;
; --- rotina de movimento das estrelas
;

MovStar:  LD  IY, yOFstar
       LD  IX, xOFstar
       LD  B, 43
Star1:    PUSH   BC
       LD  B, (IY)
       LD  C, (IX)
       PUSH   IX
       PUSH   IY
       LD  DE, blk
       CALL   WriteXY
       POP IY
       POP IX
       LD  A, (IY)
       ADD A, 01h
       DAA
       CP  23h
       JR  NZ, Star2
       XOR A
Star2:    LD  (IY), A
       LD  B, A
       LD  C, (IX)
       LD  DE, star
       PUSH   IX
       PUSH   IY
       CALL   WriteXY
       POP IY
       POP IX
       INC IX
       INC IY
       POP BC
       DJNZ   Star1
       RET
```

```
; ------------------------------------------------------------------------
;                                S I L O N
; ------------------------------------------------------------------------

;
; --- rotina de inicializacao das naves inimigas Silon
;

IniSilon:  LD  IX, silonX
       LD  IY, xOFsilon
       LD  B, 16
ISil1:     LD  A, (IX)
       LD  (IY), A
       INC IX
       INC IY
       DJNZ   ISil1
       LD  IX, silonY
       LD  IY, yOFsilon
       LD  B, 16
ISil2:     LD  A, (IX)
       LD  (IY), A
       INC IX
       INC IY
       DJNZ   ISil2
       LD  IX, silStat
       LD  B, 16
       XOR A
ISil4:     LD  (IX), A
       INC IX
       DJNZ   ISil4
       LD  IX, xOFsilon
       LD  IY, yOFsilon
       LD  B, 16
ISil3:     PUSH   BC
       LD  B, (IY)
       LD  C, (IX)
       LD  DE, silon0
       PUSH   IX
       PUSH   IY
       CALL   WriteXY
       POP IY
       POP IX
       INC IX
       INC IY
       POP BC
       DJNZ   ISil3
       RET

;
; --- rotina de movimento dos silons
;

MovSilon:  LD  IY, yOFsilon
       LD  IX, xOFsilon
       LD  HL, silStat
       LD  B, 16
MSil1:     LD  A, (HL)
       CP  DEAD
       JR  Z, MSil3
```

```
       JR  CnvSilon
MSil3:     INC IX
       INC IY
       INC HL
       DJNZ   MSil1
       RET

CnvSilon:  CALL   ClrImage
       LD  A, (cnvD1)
       CP  LEFT
       CALL   Z, CnvLeft
       CP  RIGHT
       CALL   Z, CnvRight
       CALL   PutSilon
       JR  MSil3

PutSilon:  PUSH   BC
       PUSH   IX
       PUSH   IY
       LD  B, (IY)
       LD  C, (IX)
       LD  A, (cnvImg)
       CP  IMAGE0
       JR  NZ, PSil1
       LD  DE, silon0
       JR  PSil2
PSil1:     LD  DE, silon1
PSil2:     CALL   WriteXY
       POP IY
       POP IX
       POP BC
       RET

; ------------------------------------------------------------------------------
;                                  G O R A K
; ------------------------------------------------------------------------------

;
; --- rotina de inicializacao das naves inimigas Gorak
;

IniGorak:  LD  IX, gorakX
       LD  IY, xOFgorak
       LD  B, 20
IGor1:     LD  A, (IX)
       LD  (IY), A
       INC IX
       INC IY
       DJNZ   IGor1
       LD  IX, gorakY
       LD  IY, yOFgorak
       LD  B, 20
IGor2:     LD  A, (IX)
       LD  (IY), A
       INC IX
       INC IY
       DJNZ   IGor2
       LD  IX, gorStat
       LD  B, 20
```

```
       XOR A
IGor4:     LD  (IX), A
       INC IX
       DJNZ   IGor4
       LD  IX, xOFgorak
       LD  IY, yOFgorak
       LD  B, 20
IGor3:     PUSH   BC
       LD  B, (IY)
       LD  C, (IX)
       LD  DE, gorak0
       PUSH   IX
       PUSH   IY
       CALL   WriteXY
       POP IY
       POP IX
       INC IX
       INC IY
       POP BC
       DJNZ   IGor3
       RET

;
; --- rotina de movimento dos gorak
;

MovGorak:  LD  IY, yOFgorak
       LD  IX, xOFgorak
       LD  HL, gorStat
       LD  B, 20
MGor1:     LD  A, (HL)
       CP  DEAD
       JR  Z, MGor3
       JR  CnvGorak
MGor3:     INC IX
       INC IY
       INC HL
       DJNZ   MGor1
       RET

CnvGorak:  CALL   ClrImage
       LD  A, (cnvD1)
       CP  LEFT
       CALL   Z, CnvLeft
       CP  RIGHT
       CALL   Z, CnvRight
       CALL   PutGorak
       JR  MGor3

PutGorak:  PUSH   BC
       PUSH   IX
       PUSH   IY
       LD  B, (IY)
       LD  C, (IX)
       LD  A, (cnvImg)
       CP  IMAGE0
       JR  NZ, PGor1
       LD  DE, gorak0
       JR  PGor2
```

```
PGor1:     LD  DE, gorak1
PGor2:     CALL   WriteXY
       POP IY
       POP IX
       POP BC
       RET

; ---------------------------------------------------------------------------
;                          ROTINAS GENERICAS DE MOVIMENTO
; ---------------------------------------------------------------------------

CnvLeft:   PUSH   AF
       LD  A, (IX)
       SUB 01h
       DAA
       LD  (IX), A
       CP  00h
       JR  NZ, CLeft1
       LD  A, RIGHT
       LD  (cnvD2), A
CLeft1:        POP AF
       RET

CnvRight:  PUSH   AF
       LD  A, (IX)
       ADD A, 01h
       DAA
       LD  (IX), A
       CP  78h
       JR  NZ, CRight1
       LD  A, LEFT
       LD  (cnvD2), A
CRight1:   POP AF
       RET

ClrImage:  PUSH   BC
       PUSH   IX
       PUSH   IY
       LD  B, (IY)
       LD  C, (IX)
       LD  DE, blank
       CALL   WriteXY
       POP IY
       POP IX
       POP BC
       RET

; ---------------------------------------------------------------------------
;                                    B O M B
; ---------------------------------------------------------------------------

;
; --- Rotina de inicializacao das bombas
;

IniBomb:   LD  IX, bombStat
       LD  A, FIREOFF
       LD  B, 10
IBomb1:        LD  (IX), A
```

```
        INC IX
        DJNZ   IBomb1
        RET

;
; --- Rotina de movimentacao das bombas
;

MovBomb:  LD  IX, xOFbomb
        LD  B, 10
MBomb1:       PUSH   BC
        LD  A, (IX + 20)
        CP  FIREOFF
        JR  Z, DispBomb
        CALL   ClrBomb
        LD  A, (IX + 10)
        ADD A, 01h
        DAA
        LD  (IX + 10), A
        CP  23h
        JR  NZ, MBomb2
        LD  A, FIREOFF
        LD  (IX + 20), A
        JR  MBomb3
MBomb2:       CALL   PutBomb
MBomb3:       INC IX
        POP BC
        DJNZ   MBomb1
        RET

DispBomb:  CALL   Random
        CP  0A0h
        JP  M, MBomb3
        LD  IY, xOFgorak
        LD  B, 20
DBomb1:       PUSH   BC
        LD  A, (IY + 40)
        CP  DEAD
        JP  Z, DBomb2
        CALL   Random
        CP  0A0h
        JP  M, DBomb2
        LD  A, FIREON
        LD  (IX + 20), A
        LD  A, (IY)
        ADD A, 01h
        DAA
        LD  (IX), A
        LD  A, (IY + 20)
        ADD A, 01h
        DAA
        LD  (IX + 10), A
DBomb2:       INC IY
        POP BC
        DJNZ   DBomb1
        LD  A, (IX + 20)
        CP  FIREON
        JR  Z, MBomb2
        LD  IY, xOFsilon
```

```
        LD  B, 16
DBomb3:         PUSH   BC
        LD  A, (IY + 32)
        CP  DEAD
        JP  Z, DBomb4
        CALL   Random
        CP  0A0h
        JP  M, DBomb4
        LD  A, FIREON
        LD  (IX + 20), A
        LD  A, (IY)
        ADD A, 01h
        DAA
        LD  (IX), A
        LD  A, (IY + 16)
        ADD A, 01h
        DAA
        LD  (IX + 10), A
DBomb4:         INC IY
        POP BC
        DJNZ   DBomb3
        JR  MBomb2

ClrBomb:   PUSH   HL
        PUSH   IX
        PUSH   IY
        LD  B, (IX + 10)
        LD  C, (IX)
        LD  DE, blk
        CALL   WriteXY
        POP IY
        POP IX
        POP HL
        RET

PutBomb:   PUSH   HL
        PUSH   IX
        PUSH   IY
        LD  B, (IX + 10)
        LD  C, (IX)
        LD  DE, bomb
        CALL   WriteXY
        POP IY
        POP IX
        POP HL
        RET

; ---------------------------------------------------------------------------
;                              E X P L O S I O N
; ---------------------------------------------------------------------------

; --- rotina de inicializacao das variaveis de controle da explosao

IniExpl:   XOR A
        LD  (explStat), A
        RET

; --- rotina de explosao das naves
```

```
MovExpl:  LD  A, (explStat)
       CP  03h
       JR  NZ, MExpl1
       SUB 01h
       LD  (explStat), A
       LD  IX, xOFexpl
       LD  B, (IX + 1)
       LD  C, (IX)
       LD  DE, expl0
       CALL   WriteXY
       CALL   MovScore
       RET
MExpl1:       CP  02h
       JR  NZ, MExpl2
       SUB 01h
       LD  (explStat), A
       LD  IX, xOFexpl
       LD  B, (IX + 1)
       LD  C, (IX)
       LD  DE, expl1
       CALL   WriteXY
       RET
MExpl2:       CP  01h
       RET NZ
       SUB 01h
       LD  (explStat), A
       LD  IX, xOFexpl
       LD  B, (IX + 1)
       LD  C, (IX)
       LD  DE, blank
       CALL   WriteXY
       RET

; ------------------------------------------------------------------------------
;                                 CRASH/GOTCHA
; ------------------------------------------------------------------------------

; --- rotina de verificacao de crash

TstCrash:  LD  IX, xOFbomb
       LD  B, 10
TCr1:     LD  A, (IX + 20)
       CP  FIREON
       JR  NZ, TCr2
       LD  A, 22h
       CP  (IX + 10)
       JR  NZ, TCr2
       LD  A, (xOFship)
       CP  (IX)
       JR  Z, TCr3
       ADD A, 01h
       DAA
       CP  (IX)
       JR  Z, TCr3
       ADD A, 01h
       DAA
       CP  (IX)
       JR  NZ, TCr2
TCr3:     LD  A, CRON
```

```
       LD  (shipCrash), A
       RET
TCr2:      INC IX
       DJNZ   TCr1
       RET

; --- rotina de verificacao de gotcha

TstFire:   LD  IY, xOFfire
       LD  A, (IY + 2)
       CP  FIREOFF
       RET Z
       LD  IX, xOFgorak
       LD  B, 20
TFire1:        LD  A, (IX + 40)
       CP  DEAD
       JR  Z, TFire2
       LD  A, (IX + 20)
       CP  (IY + 1)
       JR  NZ, TFire2
       LD  A, (IX)
       CP  (IY)
       JR  Z, TFire3
       ADD A, 01h
       DAA
       CP  (IY)
       JR  Z, TFire3
       ADD A, 01h
       DAA
       CP  (IY)
       JR  NZ, TFire2
TFire3:        LD  A, FIREOFF
       LD  (fireStat), A
       LD  A, CRON
       LD  (shipGotcha), A
       LD  A, DEAD
       LD  (IX + 40), A
       LD  A, (IX)
       LD  (xOFexpl), A
       LD  A, (IX + 20)
       LD  (yOFexpl), A
       LD  A, 03h
       LD  (explStat), A
       RET
TFire2:        INC IX
       DJNZ   TFire1

       LD  IX, xOFsilon
       LD  B, 16
TFire4:        LD  A, (IX + 32)
       CP  DEAD
       JR  Z, TFire6
       LD  A, (IX + 16)
       CP  (IY + 1)
       JR  NZ, TFire6
       LD  A, (IX)
       CP  (IY)
       JR  Z, TFire5
       ADD A, 01h
```

```
       DAA
       CP  (IY)
       JR  Z, TFire5
       ADD A, 01h
       DAA
       CP  (IY)
       JR  NZ, TFire6
TFire5:       LD  A, FIREOFF
       LD  (fireStat), A
       LD  A, CRON
       LD  (shipGotcha), A
       LD  A, DEAD
       LD  (IX + 32), A
       LD  A, (IX)
       LD  (xOFexpl), A
       LD  A, (IX + 16)
       LD  (yOFexpl), A
       LD  A, 03h
       LD  (explStat), A
       RET
TFire6:       INC IX
       DJNZ   TFire4
       RET

; ---------------------------------------------------------------------------
;                           TESTA FINAL DA FASE
; ---------------------------------------------------------------------------

TstFase:  LD  IX, gorStat
       LD  B, 20
TFase1:       LD  A, (IX)
       CP  DEAD
       RET NZ
       INC IX
       DJNZ   TFase1
       LD  IX, silStat
       LD  B, 16
TFase2:       LD  A, (IX)
       CP  DEAD
       RET NZ
       INC IX
       DJNZ   TFase2
       LD  A, 0FFh
       LD  (finishFase), A
       RET

; ---------------------------------------------------------------------------
;                                S C O R E
; ---------------------------------------------------------------------------

; --- rotina de inicializacao do score

IniScore:  LD  IX, score
       LD  IY, msgScore
       XOR A
       LD  (IX), A
       LD  (IX + 1), A
       LD  (IX + 2), A
       LD  A, (IX)
```

```
       CALL   Hex7h
       LD  (IY + 21), B
       LD  (IY + 22), C
       LD  A, (IX + 1)
       CALL   Hex7h
       LD  (IY + 18), B
       LD  (IY + 20), C
       LD  A, (IX + 2)
       CALL   Hex7h
       LD  (IY + 16), B
       LD  (IY + 17), C
       LD  C, CRTOUT
       LD  DE, msgScore
       CALL   BDOS
       RET

; --- rotina de incremento do score

MovScore:  LD  IX, score
       LD  IY, msgScore
       LD  A, (IX)
       ADD A, 50h
       DAA
       LD  (IX), A
       CP  00h
       JR  NZ, MScor1
       LD  A, (IX + 1)
       ADD A, 01h
       DAA
       LD  (IX + 1), A
       CP  00h
       JR  NZ, MScor1
       LD  A, (IX + 2)
       ADD A, 01h
       DAA
       LD  (IX + 2), A
MScor1:       LD  A, (IX)
       CALL   Hex7h
       LD  (IY + 21), B
       LD  (IY + 22), C
       LD  A, (IX + 1)
       CALL   Hex7h
       LD  (IY + 18), B
       LD  (IY + 20), C
       LD  A, (IX + 2)
       CALL   Hex7h
       LD  (IY + 16), B
       LD  (IY + 17), C
       LD  C, CRTOUT
       LD  DE, msgScore
       CALL   BDOS
       RET

; ------------------------------------------------------------------------------
;                                L I F E
; ------------------------------------------------------------------------------

; --- rotina de inicializacao da life
```

```
IniLife:  LD  IX, shipLife
       LD  IY, msgLife
       XOR A
       LD  (IX), A
       CALL   Hex7h
       LD  (IY + 14), B
       LD  (IY + 15), C
       LD  C, CRTOUT
       LD  DE, msgLife
       CALL   BDOS
       RET

; --- rotina de incremento do score

MovLife:  LD  IX, shipLife
       LD  IY, msgLife
       LD  A, (IX)
       SUB 01h
       DAA
       LD  (IX), A
       CALL   Hex7h
       LD  (IY + 14), B
       LD  (IY + 15), C
       LD  C, CRTOUT
       LD  DE, msgScore
       CALL   BDOS
       RET

; ------------------------------------------------------------------------------

;
; --- rotina de ajuste do tempo de espera do jogo
;

Delay:    LD  DE, TIMEC
Delay1:       LD  B, 0FFh
Delay2:       DEC B
       JR  NZ, Delay2
       DEC DE
       JR  NZ, Delay1
       RET

;
; --- rotina geradora de numero randomico
;

Random:       LD  HL, (rand1)
       PUSH   HL
       POP DE
       LD  B, 0FCh
Rnd1:     ADD HL, DE
       DJNZ   Rnd1
       LD  DE, 01h
       ADD HL, DE
       LD  (rand1), HL
       LD  A, L
       RET

;
```

```
; --- rotina de ajuste do cursor e impressao de string no video
;

WriteXY:  LD  IX, buffDisp
       LD  A, B
       PUSH   BC
       CALL   Hex7h
       LD  (IX + 2), B
       LD  (IX + 3), C
       POP BC
       LD  A, C
       PUSH   BC
       CALL   Hex7h
       LD  (IX + 5), B
       LD  (IX + 6), C
       POP BC
Write1:       LD  A, (DE)
       LD  (IX + 8), A
       INC IX
       INC DE
       CP  '$'
       JR  NZ, Write1
       LD  C, CRTOUT
       LD  DE, buffDisp
       CALL   BDOS
       RET

;
; --- rotina de conversao entre notacao BCD e ASCII
;

Hex7h:    PUSH   AF
       SRL A
       SRL A
       SRL A
       SRL A
       ADD A, 30h
       LD  B, A
       POP AF
       AND 00001111b
       ADD A, 30h
       LD  C, A
       RET

;
; --- rotina p/leitura do teclado sem buffer
;

GetKey:       LD  C, CRTOUT
       LD  DE, curOff
       CALL   BDOS
       LD  C, CONIO
       LD  E, 0FFh
       CALL   BDOS
       RET

;
; --- tabela de variaveis do programa
;
```

```
       ORG 0A00h

rand1:     DW  0000h
score:     DS  3

xOFship:   DS  1
shipLife:  DS  1
shipCrash: DS  1
shipGotcha:   DS  1
finishFase:   DS  1

xOFfire:   DS  1
yOFfire:   DS  1
fireStat:  DS  1

xOFstar:   DS  44
yOFstar:   DS  44

xOFsilon:  DS  16
yOFsilon:  DS  16
silStat:   DS  16

xOFgorak:  DS  20
yOFgorak:  DS  20
gorStat:   DS  20

xOFbomb:   DS  10
yOFbomb:   DS  10
bombStat:  DS  10

xOFexpl:   DS  1
yOFexpl:   DS  1
explStat:  DS  1

cnvD1:     DS  1
cnvD2:     DB  00h

cnvImg:       DS  1

;
; --- tabela de dados de inicializacao das variaveis
;

starX:     DB  21h, 44h
       DB  14h, 68h
       DB  27h, 59h
       DB  30h, 51h
       DB  27h, 59h
       DB  05h, 45h
       DB  22h, 68h
       DB  32h, 53h
       DB  08h, 66h
       DB  34h, 56h
       DB  18h, 67h
       DB  30h, 74h
       DB  10h, 47h
       DB  35h, 62h
       DB  25h, 72h
```

```
        DB  07h, 54h
        DB  19h, 42h
        DB  12h, 65h
        DB  36h, 73h
        DB  16h, 55h
        DB  07h, 65h
        DB  29h, 47h

starY:      DB  01h, 01h
        DB  02h, 02h
        DB  03h, 03h
        DB  04h, 04h
        DB  05h, 05h
        DB  06h, 06h
        DB  07h, 07h
        DB  08h, 08h
        DB  09h, 09h
        DB  10h, 10h
        DB  11h, 11h
        DB  12h, 12h
        DB  13h, 13h
        DB  14h, 14h
        DB  15h, 15h
        DB  16h, 16h
        DB  17h, 17h
        DB  18h, 18h
        DB  19h, 19h
        DB  20h, 20h
        DB  21h, 21h
        DB  22h, 22h

silonX:         DB      18h, 23h, 28h, 33h, 38h, 43h, 48h, 53h
        DB      18h, 23h, 28h, 33h, 38h, 43h, 48h, 53h

silonY:         DB      03h, 03h, 03h, 03h, 03h, 03h, 03h, 03h
        DB      05h, 05h, 05h, 05h, 05h, 05h, 05h, 05h

gorakX:         DB  13h, 18h, 23h, 28h, 33h, 38h, 43h, 48h, 53h, 58h
        DB  13h, 18h, 23h, 28h, 33h, 38h, 43h, 48h, 53h, 58h

gorakY:         DB  07h, 07h, 07h, 07h, 07h, 07h, 07h, 07h, 07h, 07h
        DB  09h, 09h, 09h, 09h, 09h, 09h, 09h, 09h, 09h, 09h

;
; --- tabela de dados dos objetos graficos
;

        ORG 0D00h

blank:      DB  020h, 020h, 020h, '$'

ship:       DB  0D5h, 0CFh, 0B8h, '$'
fire:       DB  0B3h, '$'

silon0:         DB  0D4h, 0CAh, 0B8h, '$'
silon1:         DB  0D5h, 0CAh, 0BEh, '$'

gorak0:         DB  0D3h, 0C2h, 0B7h, '$'
gorak1:         DB  0D6h, 0C2h, 0BDh, '$'
```

```
expl0:    DB  0B3h, 0C4h, 0B3h, '$'
expl1:    DB  0C4h, 0B3h, 0C4h, '$'

bomb:     DB  0ADh, '$'
star:     DB  02Eh, '$'

blk:      DB  020h, '$'

logo:     DB  01Bh, '[9;26H'
       DB  020h, 020h, 0D6h, 0C4h, 0C4h, 0C4h, 020h
       DB  0D6h, 0C4h, 0C4h, 0B7h, 020h, 0B7h, 020h
       DB  020h, 020h, 020h, 0D6h, 0C4h, 0C4h, 0B7h
       DB  020h, 0C4h, 0B7h, 020h, 0D6h, 020h, 020h

       DB  01Bh, '[10;26H'
       DB  020h, 020h, 0BAh, 020h, 0C4h, 0B7h, 020h
       DB  0C7h, 0C4h, 0C4h, 0B6h, 020h, 0BAh, 020h
       DB  020h, 020h, 020h, 0C7h, 0C4h, 0C4h, 0B6h
       DB  020h, 0D6h, 0C4h, 0C4h, 0BDh, 020h, 020h

       DB  01Bh, '[11;26H'
       DB  020h, 020h, 0D3h, 0C4h, 0C4h, 0BDh, 020h
       DB  0BDh, 020h, 020h, 0D3h, 020h, 0D3h, 0C4h
       DB  0C4h, 0D9h, 020h, 0BDh, 020h, 020h, 0D3h
       DB  020h, 0BDh, 020h, 0D3h, 0C4h, 020h, 020h

       DB  01Bh, '[12;26H'
       DB  0D5h, 0CDh, 0CDh, 0CDh, 0CDh, 0CDh, 0CDh
       DB  0CDh, 0CDh, 0CDh, 0CDh, 0CDh, 0CDh, 0CDh
       DB  0CDh, 0CDh, 0CDh, 0CDh, 0CDh, 0CDh, 0CDh
       DB  0CDh, 0CDh, 0CDh, 0CDh, 0CDh, 0CDh, 0BEh, '$'

base:     DB  01Bh, '[23;1H'
       DB  0DAh, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h
       DB  0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h
       DB  0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h
       DB  0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h
       DB  0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h
       DB  0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h
       DB  0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h
       DB  0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h
       DB  0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0C4h, 0BFh

       DB  0B3h, 020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h
       DB  020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h
       DB  020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h
       DB  020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h
       DB  020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h
       DB  020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h
       DB  020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h
       DB  020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h, 020h
       DB  020h, 020h, 020h, 020h, 020h, 020h, 020h, 0B3h, '$'

clear:    DB  01Bh, '[2J', '$'
curOff:      DB  01Bh, '[25;80H', '$'

msgLife:  DB  01Bh, '[24;3H', 'LIFE = 00', '$'
msgScore: DB  01Bh, '[24;64H', 'SCORE = 000', 027h, '000', '$'
```

```
msgOver:   DB  01Bh, '[12;28H', '--- G A M E   O V E R ---', '$'
msgAnyKey: DB  01Bh, '[23;14H', '--- P R E S S   A N Y   K E Y   T O   B E G I N
---', '$'

;
; --- declaracao da area de buffer do display
;

       ORG 1C00h

buffDisp:  DB  01Bh, '[00;00H'
       DS  0100h

;
; --- declaracao da area de stack
;

       ORG 1F00h

stack:     DS  0100h

       END
   ]]</CEPEL_Data>
<Signature
xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/><SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/><Reference
URI="#202309101000376"><Transforms><Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/><Transform
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/></Transforms><DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/><DigestValue>CMKyIi03WYo4DTGX4M
+4XVKG8Ks=</DigestValue></Reference></SignedInfo><SignatureValue>iHwxwPGQpt/tgspDLF
l/AIXwg4Xs0xDTXbYmVcY3ATGOHi6uiwZilrAlS5lVzFWel+MS98YX0p3C
mOkQl85zZlxh3xJGGkcOhR4q5UXPjFDmm62Q5II0bsE8T9XvDA45gfs1mPessqyNmisycGFcYy1g
cRZcB6xO+1GJ4U3wOKBij7AyIr52nL/5HGM7SuGgelRTD5zX3SuiNeXIiFKxXYVgf1YRN89FF8Oc
XePrVEd4Hqgp1arXYR0DoYk4VABRtxYRqmV3e5NA+x+32//FoUO8DoAzrrptaPug9tSTrwqQvDP/
IyZxN8FvIoq4ZKNfk8BZAace3Dit5ZE3wjwZDg==</SignatureValue><KeyInfo><X509Data><X509Ce
rtificate>MIIIBTCCBe2gAwIBAgIIWrHgZiERClswDQYJKoZIhvcNAQELBQAwdTELMAkGA1UEBhMCQlIxE
zAR
BgNVBAoMCklDUC1CcmFzaWwxNjA0BgNVBAsMLVNlY3JldGFyaWEgZGEgUmVjZWl0YSBGZWRlcmFs
IGRvIEJyYXNpbCAtIFJGQjEZMBcGA1UEAwwQQUMgU0VSQVNBIFJGQiB2NTAeFw0yMjA2MDIxMDI3
MDBaFw0yMzA2MDIxMDI3MDBaMIIBFDELMAkGA1UEBhMCQlIxCzAJBgNVBAgMAlNQMR4wHAYDVQQH
DBVTYW8gQmVybmFyZG8gZG8gQ2FtcG8xEzARBgNVBAoMCklDUC1CcmFzaWwxNjA0BgNVBAsMLVNl
Y3JldGFyaWEgZGEgUmVjZWl0YSBGZWRlcmFsIGRvIEJyYXNpbCAtIFJGQjEWMBQGA1UECwwNUkZC
IGUtQ05QSiBBMTEWMBQGA1UECwwNQUMgU0VSQVNBIFJGQjEXMBUGA1UECwwOMTQxODMzMjQwMDAx
MTcxEzARBgNVBAsMClBSRVNFTkNJQUwxLTArBgNVBAMMJFRBVVJJIEVOR0VOSEFSSUEgTFREQToy
MzU4NzU0MjAwMDEwODCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAM+2xi63F01fJ58O
V3zqAq4TAbiA+QSz5oTmSM+tL7YAn9yIfYoZ4TXInRWSQmFD496DMjc/0dxTA+noO8Ddolyt3xw7
OfJz2eMlF3E5L7XghV3V4dOSvef4hyvdJqij6+4jGMCu2Dkos9Ioq+TNqeaLpI4vY5V+Ev7CNrZe
UhqY9A5a/j5ENCCXG/2PVarwH4Y1GuhhpMxueifTbrvfhouDLKLc9px3htYXBFYhPzN1bUaMqQhi
C3jiJuOdc3SQC2KaieTUbyZYiAeu9sjfGfwW1Oc6XRnmEe4BhNM2GSA7nKDis0aVobwBkoCWRY1c
x/ZXd589Rr+pTq+2F41QYuECAwEAAaOCAvYwggLyMAkGA1UdEwQCMAAwHwYDVR0jBBgwFoAU7PFB
UVeo5jrpXrOgIvkIirU6h48wgZkGCCsGAQUFBwEBBIGMMIGJMEgGCCsGAQUFBzAChjxodHRwOi8v
d3d3LmNlcnRpZmljYWRvZGlnaXRhbC5jb20uYnIvY2FkZWlhcy9zZXJhc2FyZmJ2NS5wN2IwPQYI
KwYBBQUHMAGGMWh0dHA6Ly9vY3NwLmNlcnRpZmljYWRvZGlnaXRhbC5jb20uYnIvc2VyYXNhcmZi
djUwgcYGA1UdEQSBvjCBu4EYRklOQU5DRUlST0BUQVVSSS1FTkcuQ09NoCsGBWBMAQMCoCITIE1J
Q0hBRUwgRU5SSVFVRSBWRUdBIExFT04gU0VJSkFToBkGBWBMAQMDoBATDjIzNTg3NTQyMDAwMTA4
oD4GBWBMAQMEoDUTMzIxMDQxOTg2MjMxODAxMDc4MTIwMDAwMDAwMDAwMDAwMDAwMDAwMDAw
```

MDAwMDAwMKAXBgVgTAEDB6AOEwwwMDAwMDAwMDAwMDAwcQYDVR0gBGowaDBmBgZgTAECAQ0wXDBa
BggrBgEFBQcCARZOaHR0cDovL3B1YmxpY2FjYW8uY2VydGlmaWNhZG9kaWdpdGFsLmNvbS5ici9y
ZXBvc2l0b3Jpby9kcGMvZGVjbGFyYWNhby1yZmIucGRmMB0GA1UdJQQWMBQGCCsGAQUFBwMCBggr
BgEFBQcDBDCBnQYDVR0fBIGVMIGSMEqgSKBGhkRodHRwOi8vd3d3LmNlcnRpZmljYWRvZGlnaXRh
bC5jb20uYnIvcmVwb3NpdG9yaW8vbGNyL3NlcmFzYXJmYnY1LmNybDBEoEKgQIY+aHR0cDovL2xj
ci5jZXJ0aWZpY2Fkb3MuY29tLmJyL3JlcG9zaXRvcmlvL2xjci9zZXJhc2FyZmJ2NS5jcmwwHQYD
VR0OBBYEFPZiuqzHIKdFieU4dhgELgU0N536MA4GA1UdDwEB/wQEAwIF4DANBgkqhkiG9w0BAQsF
AAOCAgEAGmlYia37FmMDrgnKsebQVi3WbKRJB7o/xDEFsxuhW9lsntB0NHyEzg2IwDrdTSaJ9A1t
M21nWT6a7f217ML1HeoXIxBieyQRYeK1zlVVO4SbDyfZogwWnG2pBeu4A/NkS49H6eiPp5DMinrS
lfNHMTWCrEn5GzMnQPn+xzTSLB/KDgAg48/4L/JtwbUo6DUuz34glKiOB/AG1wlUvthuOIhMubYw
iAMnOPOhBKaJTqZuAB+E14niV1sG1xSVwxPvIsVGLN8oq9gRfxMOT9mzmk0exgHZd4yfzT72DZc0
sXPiTQ1W8t7oy9j/UpG9o0OonuawnBQP4rmoIvsglRTwPtn/uGHhQC3Hy1QpIx990XTS6aNsWSU+
3Ed7phBgJQl1Ozj3lqpAyM4oofnzP8gxqdncS8ozxTYO0GYEGT1yBCMt7AzQjS8GtCNvzB0hZ3fy
/i35CdX6QNwRqnGO8EosqP+mNV8y7N4AJX/StyInlAL+VRqsCqHC8lz6IynZ95aGlI0wW0gEMC/h
WRY5xiV/7Ve/9GmKR5xSkdtJ0ZXv9WMn7oAxuoo9tylPjcKfYCkRy/dGhCo00ykmy0v7qoSvmoai
IEt67leVRs1yRoRXCWj85gbnkCbJlzQ854XvLWAYwV7/1rWo5bhrQEKgoSepFOCy6L2FuLsgAGSJ
NGt6Y+o=</X509Certificate></X509Data></KeyInfo></Signature></CEPEL_GEDWEB>