

```

Bank1:72B2 ; Konami logo logic
Bank1:72B2 ;
Bank1:72B2 ;-----
Bank1:72B2
Bank1:72B2 GS_KonamiLogo:
Bank1:72B2     ld     a, b
Bank1:72B3     cp     1
Bank1:72B5     jr     z, GS_KonamiLogo2
Bank1:72B7
Bank1:72B7     jr     nc, GS_KonamiLogo3
Bank1:72B9
Bank1:72B9     call    InitScreen
Bank1:72BC     call    SetBanks_D_E
Bank1:72BF     call    DrawKonamiLogo
Bank1:72C2     call    SetBanks_1_2_3
Bank1:72C5     jp     NextSubstatus
Bank1:72C8
Bank1:72C8 GS_KonamiLogo2:
Bank1:72C8     call    SetBanks_D_E
Bank1:72CB     call    DrawLogoEffect
Bank1:72CE     call    SetBanks_1_2_3
Bank1:72D1
Bank1:72D1     ld     a, (LogoFinishedFlag)
Bank1:72D4     or     a
Bank1:72D5     ret    z
Bank1:72D6
Bank1:72D6     call    LoadFontsGfx
Bank1:72D9     xor    a
Bank1:72DA     jr     NextSubstatusT
Bank1:72DC
Bank1:72DC GS_KonamiLogo3:
Bank1:72DC     ld     hl, Timer
Bank1:72DF     dec    (hl)
Bank1:72E0     ret    nz
Bank1:72E1
Bank1:72E1     call    InitScreen
Bank1:72E4
Bank1:72E4     ld     bc, 7
Bank1:72E7     call    WRTUDP
Bank1:72EA
Bank1:72EA     call    DisableSprites
Bank1:72ED     call    GS_KonamiLogo5
Bank1:72F0     call    EraseSprAttribRAM
Bank1:72F3
Bank1:72F3     call    InitSprAttribUram
Bank1:72F6     call    EnableSprites
Bank1:72F9
Bank1:72F9     ld     a, (IntroSelector)
Bank1:72FC     and    a
Bank1:72FD     ld     hl, 1
Bank1:7300     jr     z, GS_KonamiLogo4
Bank1:7302
Bank1:7302     ld     hl, 80h
Bank1:7305
Bank1:7305 GS_KonamiLogo4:
Bank1:7305     ld     a, (OpeningStatus), hl
Bank1:7308     ld     hl, (IntroSelector), hl
Bank1:7309
Bank1:7309     ld     a, (hl)
Bank1:730C     xor    1

```



Mirando detrás de los píxeles

**¿EN  
QUÉ  
CONSISTE  
DESENSAMBLAR  
UN JUEGO?**

# EL Z80 SÓLO ENTIENDE CÓDIGO MÁQUINA

El procesador solo entiende una serie de instrucciones que le indican qué acciones tiene que realizar.



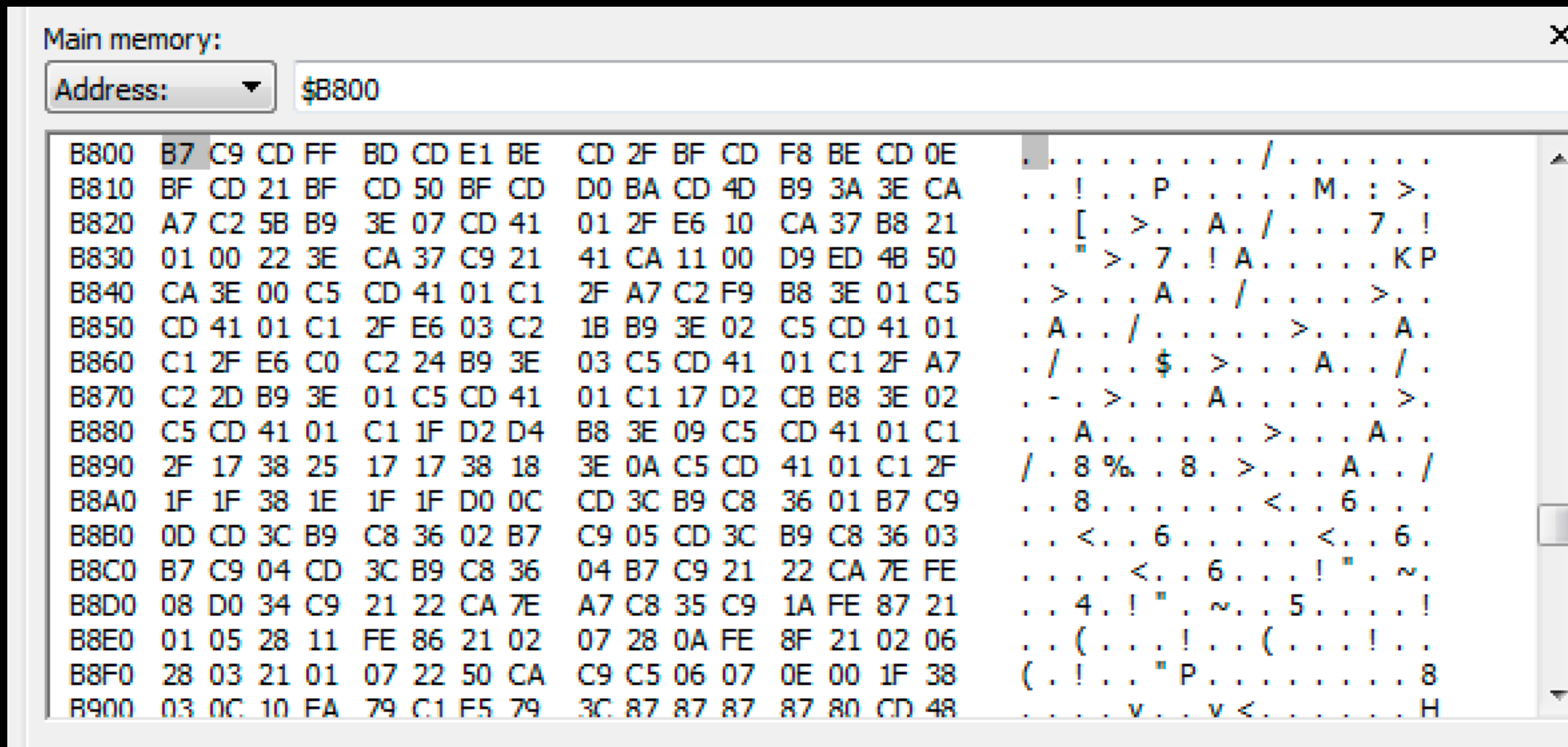
**00111111**

**Suma 2+2**



# ENSAMBLADOR

El código máquina es muy difícil para un humano. El lenguaje ensamblador implementa una representación simbólica de los códigos de máquina binarios.



The screenshot shows a debugger window titled "Main memory:" with a close button (X) in the top right corner. Below the title bar, there is a label "Address:" followed by a dropdown menu and the value "\$B800". The main area of the window displays a list of memory addresses from B800 to B900. Each address is followed by a column of 16 hexadecimal machine code bytes, and then a column of their corresponding ASCII characters. The ASCII characters are often truncated with dots (e.g., "...").

Address	Machine Code (Hex)	ASCII
B800	B7 C9 CD FF BD CD E1 BE CD 2F BF CD F8 BE CD 0E	..... / .....
B810	BF CD 21 BF CD 50 BF CD D0 BA CD 4D B9 3A 3E CA	.. ! .. P .. .. M. : >.
B820	A7 C2 5B B9 3E 07 CD 41 01 2F E6 10 CA 37 B8 21	.. [ . >.. A. / .. .. 7. !
B830	01 00 22 3E CA 37 C9 21 41 CA 11 00 D9 ED 4B 50	.. " >. 7. ! A .. .. KP
B840	CA 3E 00 C5 CD 41 01 C1 2F A7 C2 F9 B8 3E 01 C5	. >.. .. A.. / .. .. >..
B850	CD 41 01 C1 2F E6 03 C2 1B B9 3E 02 C5 CD 41 01	. A.. / .. .. >.. .. A.
B860	C1 2F E6 C0 C2 24 B9 3E 03 C5 CD 41 01 C1 2F A7	. / .. .. \$ . >.. .. A.. / .
B870	C2 2D B9 3E 01 C5 CD 41 01 C1 17 D2 CB B8 3E 02	.. - . >.. .. A .. .. >.
B880	C5 CD 41 01 C1 1F D2 D4 B8 3E 09 C5 CD 41 01 C1	.. A .. .. >.. .. A..
B890	2F 17 38 25 17 17 38 18 3E 0A C5 CD 41 01 C1 2F	/ . 8 % . 8 . >.. .. A.. /
B8A0	1F 1F 38 1E 1F 1F D0 0C CD 3C B9 C8 36 01 B7 C9	.. 8 .. .. <.. .. 6 ..
B8B0	0D CD 3C B9 C8 36 02 B7 C9 05 CD 3C B9 C8 36 03	.. <.. .. 6 .. .. <.. .. 6.
B8C0	B7 C9 04 CD 3C B9 C8 36 04 B7 C9 21 22 CA 7E FE	.. .. <.. .. 6 .. .. ! " . ~.
B8D0	08 D0 34 C9 21 22 CA 7E A7 C8 35 C9 1A FE 87 21	.. 4. ! " . ~. . 5 .. .. !
B8E0	01 05 28 11 FE 86 21 02 07 28 0A FE 8F 21 02 06	.. ( .. .. ! .. ( .. .. ! ..
B8F0	28 03 21 01 07 22 50 CA C9 C5 06 07 0E 00 1F 38	( . ! .. " P .. .. .. 8
B900	03 0C 10 FA 79 C1 F5 79 3C 87 87 87 87 80 CD 48	.. .. v .. .. v < .. .. .. H



# ENSAMBLADOR

El código máquina es muy difícil para un humano. El lenguaje ensamblador implementa una representación simbólica de los códigos de máquina binarios.

Main memory:

Address:

B800	B7	C9	CD	FF	BD	CD	E1	BE	CD	2F	BF	CD	F8	BE	CD	0E
B810	BF	CD	21	BF	CD	50	BF	CD	D0	BA	CD	4D	B9	3A	3E	CA
B820	A7	C2	5B	B9	3E	07	CD	41	01	2F	E6	10	CA	37	B8	21
B830	01	00	22	3E	CA	37	C9	21	41	CA	11	00	D9	ED	4B	50
B840	CA	3E	00	C5	CD	41	01	C1	2F	A7	C2	F9	B8	3E	01	C5
B850	CD	41	01	C1	2F	E6	03	C2	1B	B9	3E	02	C5	CD	41	01
B860	C1	2F	E6	C0	C2	24	B9	3E	03	C5	CD	41	01	C1	2F	A7
B870	C2	2D	B9	3E	01	C5	CD	41	01	C1	17	D2	CB	B8	3E	02
B880	C5	CD	41	01	C1	1F	D2	D4	B8	3E	09	C5	CD	41	01	C1
B890	2F	17	38	25	17	17	38	18	3E	0A	C5	CD	41	01	C1	2F
B8A0	1F	1F	38	1E	1F	1F	D0	0C	CD	3C	B9	C8	36	01	B7	C9
B8B0	0D	CD	3C	B9	C8	36	02	B7	C9	05	CD	3C	B9	C8	36	03
B8C0	B7	C9	04	CD	3C	B9	C8	36	04	B7	C9	21	22	CA	7E	FE
B8D0	08	D0	34	C9	21	22	CA	7E	A7	C8	35	C9	1A	FE	87	21
B8E0	01	05	28	11	FE	86	21	02	07	28	0A	FE	8F	21	02	06
B8F0	28	03	21	01	07	22	50	CA	C9	C5	06	07	0E	00	1F	38
B900	03	0C	10	FA	79	C1	F5	79	3C	87	87	87	87	80	CD	48

```
Code view:
B800 B7      or      a
B801 C9      ret
B802 CD FF BD call    #bfff
B805 CD E1 BE call    #bee1
B808 CD 2F BF call    #bf2f
B80B CD F8 BE call    #bef8
B80E CD 0E BF call    #bf0e
B811 CD 21 BF call    #bf21
B814 CD 50 BF call    #bf50
B817 CD D0 BA call    #bad0
B81A CD 4D B9 call    #b94d
B81D 3A 3E CA ld      a,(#ca3e)
B820 A7      and     a
B821 C2 5B B9 jp      nz,#b95b
B824 3E 07  ld      a,#07
B826 CD 41 01 call    #0141
B829 2F      cpl
B82A E6 10  and     #10
B82C CA 37 B8 jp      z,#b837
B82F 21 01 00 ld      hl,#0001
B832 22 3E CA ld      (#ca3e),hl
B835 37      scf
B836 C9      ret
B837 21 41 CA ld      hl,#ca41
B83A 11 00 D9 ld      de,#d900
B83D ED 4B 50 CA ld      bc,(#ca50)
B841 3E 00  ld      a,#00
B843 C5      push   bc
B844 CD 41 01 call    #0141
B847 C1      pop    bc
B848 2F      cpl
B849 A7      and     a
B84A C2 F9 B8 jp      nz,#b8f9
B84D 3E 01  ld      a,#01
B84F C5      push   bc
B850 CD 41 01 call    #0141
B853 C1      pop    bc
B854 2F      cpl
B855 E6 03  and     #03
B857 C2 1B B9 jp      nz,#b91b
B85A 3E 02  ld      a,#02
B85C C5      push   bc
B85D CD 41 01 call    #0141
B860 C1      pop    bc
B861 2F      cpl
B862 E6 C0  and     #c0
```

```

B800 ;-----
B800 ;
B800 ; Debug menu
B800 ; Disabled in final ROM
B800 ;
B800 ; Out:
B800 ;   Cy = Skip game logic
B800 ;-----
B800 DEBUG:
B800         or      a
B801         ret
B802
B802         call    DbgWeaponItems      ; Z: All weapons and items
B805         call    DbgCards            ; X: All ID cards
B808         call    DbgRations          ; R: Add rations B1x20, B2x40 and B3x60
B80B         call    DbgInvisibleMode   ; Invisible mode INS: enable, DEL: disable
B80E         call    DbgJunglePath      ; BS: Snake knows the path in the jungle
B811         call    DbgTransfBrooch    ; T: Transform Natasha's brooch into Dr. Petrovich's locker key
B814         call    DbgEnding          ; =: Show the ending
B817         call    DbgSnakeXY        ; P: Print Snake coordinates
B81A         call    DbgSkipText        ; NUM.: Skip mission briefing
B81D
B81D         ld      a, (DbgStopModeF)
B820         and
B821         jp      nz, DbgStopMode2
B824
B824         ld      a, 7
B826         call    SNSMAT
B829         cpl
B82A         and    10h                ; STOP
B82C         jp      z, DbgSelectZone   ; Not pressed
B82F
B82F         ld      hl, 1
B832         ld      (DbgStopModeF), hl
B835         scf
B836         ret
B837
B837 DbgSelectZone:
B837         ld      hl, ChangeRoomDir
B83A         ld      de, RoomsMap
B83D         ld      bc, (RoomsMapY)
B841
B841         ld      a, 0                ; (??) XOR A
B843         push   bc
B844         call   SNSMAT
B847         pop    bc
B848         cpl
B849         and    a                    ; 0-7
B84A         jp      nz, DgbZone_0_7
B84D
B84D         ld      a, 1
B84F         push   bc
B850         call   SNSMAT
B853         pop    bc
B854         cpl
B855         and    3                    ; 8-9
B857         jp      nz, DgbZone_8_9
B85A
B85A         ld      a, 2
B85C         push   bc
B85D         call   SNSMAT
B860         pop    bc
B861         cpl
B862         and    0C0h                ; A-B

```

Para hacerlo más legible:

- Estudiar qué hace el código
- Renombrar variables y funciones
- Añadir comentarios

```

Code view:
B800 B7      or      a
B801 C9      ret
B802 CD FF BD  call   #bdff
B805 CD E1 BE  call   #bee1
B808 CD 2F BF  call   #bf2f
B80B CD F8 BE  call   #bef8
B80E CD 0E BF  call   #bf0e
B811 CD 21 BF  call   #bf21
B814 CD 50 BF  call   #bf50
B817 CD D0 BA  call   #bad0
B81A CD 4D B9  call   #b94d
B81D 3A 3E CA  ld     a,(#ca3e)
B820 A7      and
B821 C2 5B B9  jp     nz,#b95b
B824 3E 07    ld     a,#07
B826 CD 41 01  call   #0141
B829 2F      cpl
B82A E6 10    and   #10
B82C CA 37 B8  jp     z,#b837
B82F 21 01 00  ld     hl,#0001
B832 22 3E CA  ld     (#ca3e),hl
B835 37      scf
B836 C9      ret
B837 21 41 CA  ld     hl,#ca41
B83A 11 00 D9  ld     de,#d900
B83D ED 4B 50 CA  ld     bc,(#ca50)
B841 3E 00    ld     a,#00
B843 C5      push  bc
B844 CD 41 01  call   #0141
B847 C1      pop   bc
B848 2F      cpl
B849 A7      and   a
B84A C2 F9 B8  jp     nz,#b8f9
B84D 3E 01    ld     a,#01
B84F C5      push  bc
B850 CD 41 01  call   #0141
B853 C1      pop   bc
B854 2F      cpl
B855 E6 03    and   #03
B857 C2 1B B9  jp     nz,#b91b
B85A 3E 02    ld     a,#02
B85C C5      push  bc
B85D CD 41 01  call   #0141
B860 C1      pop   bc
B861 2F      cpl
B862 E6 C0    and   #c0

```

**COSAS  
QUE  
SE  
ENCUENTRAN  
DESENSAMBLANDO**



# EL CONTRA NO FUE PROGRAMADO POR KONAMI



Desde el Nemesis, todos los juegos en los que hay disparos utilizan la misma rutina (p.e.: Vampire Killer, King Kong 2, Solid Snake...) La única excepción es el Penguin Adventure, probablemente porque se programó al mismo tiempo. Sin embargo el Contra no la usaba, ni estaba programado como el resto de Konamis.

¿Quién lo programó entonces?

Busqué juegos que usasen rutinas similares para gestionar las interrupciones, o leer los controles. Y el sorprendente resultado fue: AshGuine, Famicle Parodic 1 y 2, Kubikiri, Nyancle, Quinpl, Tengoku Yoitoko (Taito) y Mon Mon Monster (Hot-B).

	Contra	Famicle Parodic	Tengoku Yoitoko
push	h1	push	h1
push	bc	push	bc
push	af	push	af
ld	a, 1	ld	a, 1
and	0E0h ; 'a'	and	0E0h ; 'a'
rrc	h	rrc	h
rca	h	rca	h
rca	h	rca	h
ld	(0E0D6h), a	ld	(0E0D6h), a
ld	a, h	ld	a, h
ld	(0E0D7h), a	ld	(0E0D7h), a
ld	a, 1	ld	a, 1
and	1Fh	and	1Fh
rlca		rlca	
rlca		rlca	
ld	(0E0D4h), a	ld	(0E0D4h), a
pop	af	pop	af
ld	1, a	ld	1, a
and	0E0h ; 'a'	and	0E0h ; 'a'
rrca		rrca	
rrca		rrca	
ld	h, a	ld	h, a
ld	a, (0E01Fh)	ld	a, (0E01Fh)
add	a, h	add	a, h
ld	(0E0D2h), a	ld	(0E0D2h), a
ld	a, 1	ld	a, 1
and	1Fh	and	1Fh
rlca		rlca	
rlca		rlca	
rlca		rlca	
ld	(0E0D0h), a	ld	(0E0D0h), a
ld	b, 20h ; ''	ld	b, 20h ; ''
call	loc_10107B	call	loc_100B41
pop	bc	pop	bc
pop	h1	pop	h1
ret		ret	



# PENGUIN ADVENTURE

Tiene dos finales:

- Bueno: Llegas a tiempo de salvar a la princesa.
- Malo: Llegas tarde y la princesa se muere.

¿De qué depende ver uno u otro?





Pare ver el final juego hay que pausar el juego  $(n * 4) + 1$  veces

```
Bank2:82CA      ld      a, (Stage)
Bank2:82CD      cp      12
Bank2:82CF      ld      c, 2          ; Apple tree cinematic
Bank2:82D1      jr      z, SetCinematicScene
Bank2:82D3
Bank2:82D3      cp      24          ; Last stage
Bank2:82D5      jr      z, CheckBadEnding
Bank2:82D7
Bank2:82D7 SkipCinematic: ; ...
Bank2:82D7      ld      hl, EngineSubStatus
Bank2:82DA      inc    (hl)        ; Skip cinematic scene
Bank2:82DB      jp     NextSubstatus
Bank2:82DE ; -----
Bank2:82DE //-----
Bank2:82DE // Check if the game was paused (n * 4) + 1 times (E.g.: 1, 5, 9...)
Bank2:82DE //-----
Bank2:82DE
Bank2:82DE CheckBadEnding: ; ...
Bank2:82DE      ld      a, (TimesPaused)
Bank2:82E1      and    3          ; Limit number to 0-3
Bank2:82E3      dec    a
Bank2:82E4      ld      c, 0          ; Good ending = 0
Bank2:82E6      jr      z, SetCinematicScene
Bank2:82E8
Bank2:82E8      inc    c          ; Bad ending = 1
Bank2:82E9
Bank2:82E9 SetCinematicScene: ; ...
Bank2:82E9      ld      a, c
Bank2:82EA      ld      (CinematicScene), a ; 0 = Good ending, 1 = Bad Ending, 2 = Apple tree
Bank2:82ED
```

TIME051 DIST0000  025 SP



# • MÁQUINAS TRAGAPERRAS

Por muy rápido que seas no vas a poder saber qué símbolo saldrá.



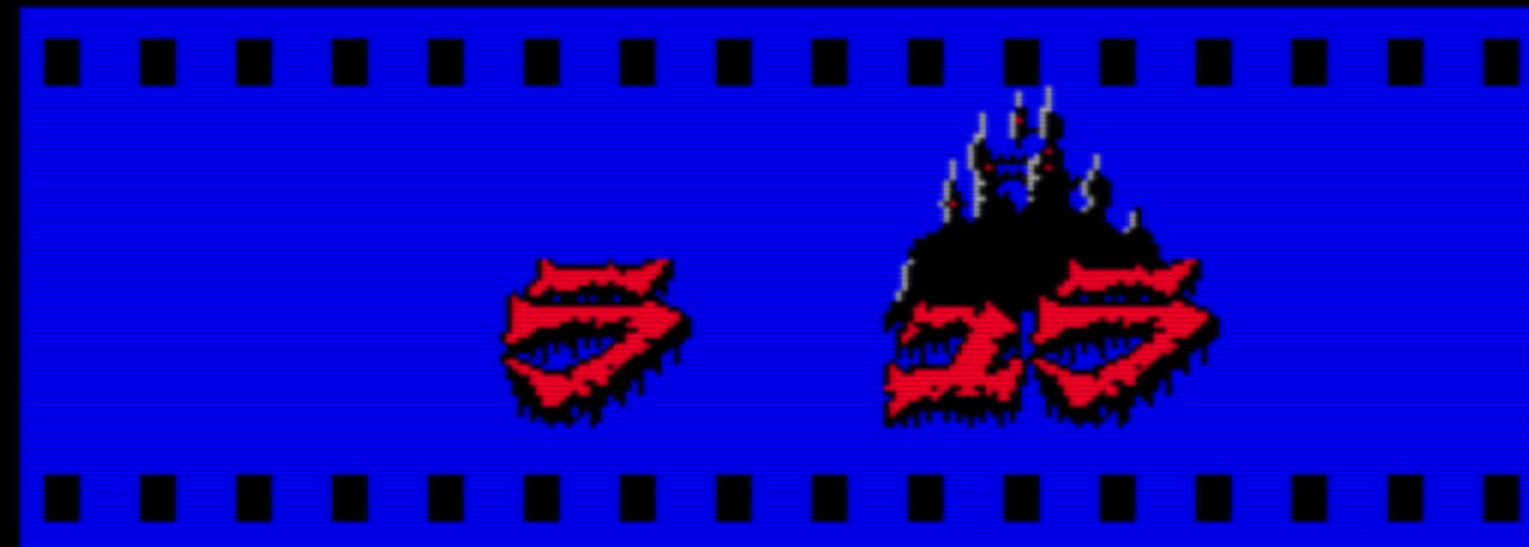
```
Bank1:7999          ld    de, CurrentSymbolC
Bank1:7999          ld    hl, 3A1Ah          ; URAM address
Bank1:799F          SlotMachSpinning2:          ; ...
Bank1:799F          push  bc
Bank1:79A0          push  hl
Bank1:79A1          ld    a, r              ; Random value
Bank1:79A3          and   0Fh              ; Limit value to 0-15
Bank1:79A5          push  hl
Bank1:79A6          ld    hl, SymbolsList
Bank1:79A9          call  ADD_HL_A
Bank1:79AC          ld    a, (hl)          ; Get symbol ID
Bank1:79AD          ld    (de), a
Bank1:79AE          ld    hl, Symbols1stTile
Bank1:79B1          call  ADD_HL_A
Bank1:79B4          ld    c, (hl)          ; Get symbol first tile number
Bank1:79B5          pop   hl
Bank1:79B6          call  DrawSymbol
Bank1:79B9          pop   hl
Bank1:79BA          dec   hl
Bank1:79BB          dec   hl
Bank1:79BC          dec   hl
Bank1:79BD          dec   hl              ; Move URAM pointer to the next symbol slot
Bank1:79BE          dec   de              ; Address to store current symbol
Bank1:79BF          pop   bc
Bank1:79BF          djnz SlotMachSpinning2
Bank1:79C0          ld    a, (ControlTrigger)
Bank1:79C2          and   10h
Bank1:79C5          ret   z              ; Fire button not pressed
Bank1:79C7          ld    a, 2Ah
Bank1:79C8          ld    hl, NumRollingReels
Bank1:79CA          dec   (hl)
Bank1:79CD          jp   nz, PlaySound_Check
Bank1:79D1
```



# AKUMAJYO DRAKYULA - VAMPIRE KILLER



El logo tiene partes hechas con sprites.  
Doki doki es una onomatopeya en japonés del latido del corazón, y se usa para indicar excitación, nerviosismo, miedo. Quizá iban a animar esas partes.



©KONAMI 1986

PUSH SPACE KEY

# • RELOJ DE ARENA

Hay un objeto "desconocido" que aparece cuando golpeas el reloj de arena.

```
Bank2:8D4F PickItem3:
Bank2:8D4F      ld      (ItemToPick), a
Bank2:8D52      call   ShowItem
Bank2:8D55      cp      CHAIN_WHIP
Bank2:8D57      jr      nc, PickWeapon
Bank2:8D59
Bank2:8D59      dec     a
Bank2:8D5A      call   JumpIndex
Bank2:8D5D
Bank2:8D5D      dw     PickSmallHeart
Bank2:8D5F      dw     PickBigHeart
Bank2:8D61      dw     PickBronzeShield
Bank2:8D63      dw     PickGoldenShield
Bank2:8D65      dw     PickCrossGold
Bank2:8D67      dw     PickCrossSilver
Bank2:8D69      dw     PickRedOrb
Bank2:8D6B      dw     PickBlueOrb
Bank2:8D6D      dw     PickBlueRing
Bank2:8D6F      dw     PickHourglass
Bank2:8D71      dw     PickHourglassRot
Bank2:8D73      dw     PickBoots
Bank2:8D75      dw     PickWings
Bank2:8D77      dw     PickCandle
Bank2:8D79      dw     PickMap
Bank2:8D7B      dw     PickBlackBible
Bank2:8D7D      dw     PickWhiteBible
Bank2:8D7F      dw     PickRod
Bank2:8D81      dw     AddPoints5K
Bank2:8D83      dw     AddPoints1K
Bank2:8D85      dw     PickDummy
Bank2:8D87      dw     PickPotion
Bank2:8D89      dw     PickKeySmall
Bank2:8D8B      dw     PickKeyBig
Bank2:8D8D      dw     PickDummy
```



# • RELOJ DE ARENA

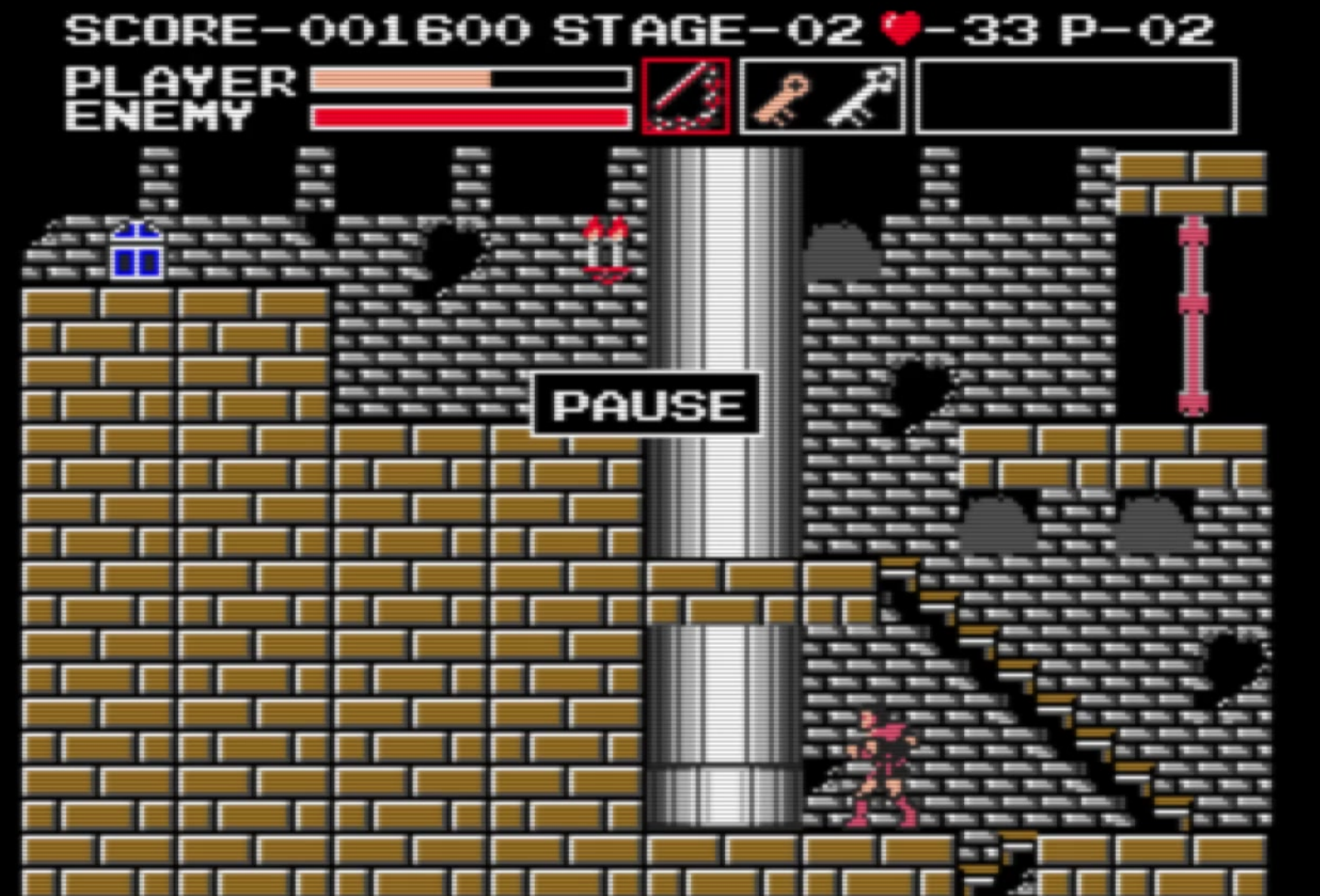
El reloj de arena tumbado hace que la duración de otros objetos sea mayor.



The image shows the title screen for '悪魔城ドラキュラ' (Dracula X) with a character holding a sword. Below the title, an illustration shows a character, an upright hourglass, and a tilted hourglass with an arrow pointing from the upright to the tilted one, indicating the arena clock mechanic.

	150	240
	150	240
	150	240
	90	150

30 = 1s aprox. at 60Hz





# • HAY PARTES QUE NO HEMOS VISTO



- Las habitaciones están formadas por 8x6 metatiles de 32x32 pixeles (256x192)
- La resolución es de 256x212.
- El marcador tapa la parte superior.



# METAL GEAR





# • OBJETOS QUE DEJAN LOS SOLDADOS

- Inicialmente varios tipos de soldados podían dejar munición o comida al darles tres puñetazos.
- Pero por un cambio de última hora, o un fallo, solo los que andan lento lo hacen.

```
Bank1:72F9 ;-----
Bank1:72F9 ;
Bank1:72F9 ; After three punches the guard dies
Bank1:72F9 ; Some of them drop an item
Bank1:72F9 ;
Bank1:72F9 ;-----
Bank1:72F9 ChkKillPunching:
Bank1:72F9         ; ---
Bank1:72F9         inc     (ix+ACTOR.PunchesCnt) ; Increment number of punches
Bank1:72FC         ld     a, (ix+ACTOR.PunchesCnt)
Bank1:72FF         cp     3
Bank1:7301         jr     z, ChkDropItem ; After 3 punches, check if the enemy drops an item
Bank1:7303         ld     (ix+ACTOR.StunnedCnt), 40h ; Stunned time
Bank1:7307         ret
Bank1:7308 ;-----
Bank1:7308 ;
Bank1:7308 ;
Bank1:7308 ; Chks if the enemy drops a ration or ammo crate
Bank1:7308 ; (??) This routine is weird. Only the first enemy type checked can drop an item
Bank1:7308 ;
Bank1:7308 ;-----
Bank1:7308 ChkDropItem:
Bank1:7308         ; ---
Bank1:7308         ld     a, (ix+ACTOR) ; Enemy type
Bank1:7308         cp     ID_GUARD_SLOW ; (??) Only this ID drops an item
Bank1:7308         jr     nz, ChkDropItem3 ; The NZ filters all other enemy types
Bank1:730F         cp     ID_GUARD_MEDIUM
Bank1:7311         jr     z, ChkDropItem2
Bank1:7313         cp     ID_GUARD_ELEVATOR
Bank1:7315         jr     z, ChkDropItem2
Bank1:7317         cp     ID_GUARD_FAST
Bank1:7319         jr     z, ChkDropItem2
Bank1:731B ; (Probably some missing code here)
Bank1:731B ChkDropItem2:
Bank1:731B         ; ---
Bank1:731B         ld     a, (ix+ACTOR.Y)
Bank1:731E         sub     4
Bank1:7320         ld     e, a
Bank1:7321         ld     a, (ix+ACTOR.X)
Bank1:7324         sub     8
Bank1:7326         ld     d, a ; DE = Item XY
Bank1:7327         ld     a, r ; Random number
Bank1:7329         rra
Bank1:732A         rra
Bank1:732B         and     3 ; 0-3
Bank1:732D         cp     2 ; Value threshold
Bank1:732F         ld     c, a ; Item ID (0 = Ratio, 1 = Ammo crate)
Bank1:7330         call    c, SpawnItem_
Bank1:7333 ChkDropItem3:
Bank1:7333         ; ---
Bank1:7333         jp     KillActor
```



# • SUBIR DE RANGO

Si intentamos rescatar al Dr. Madnar sin haber rescatado a su hija Ellen antes, se negará a ayudarnos. Pero el contador de prisioneros rescatados se incrementará igualmente (cada 5 nos suben de rango). Así que podemos repetir la operación las veces que queramos.

```
Bank2:9CF4 ;-----
Bank2:9CF4 ;
Bank2:9CF4 ; Dr. Pretrovich Madnar logic
Bank2:9CF4 ;
Bank2:9CF4 ;-----
Bank2:9CF4
Bank2:9CF4 MadnarLogic:
Bank2:9CF4         ld      a, (RescuedArray+14h) ; ...
Bank2:9CF7         or      a          ; Is Ellen rescued?
Bank2:9CF8         ld      a, 124          ; TEXT: I'M DR. PETTROVICH.*MY DAUGHTER ELLEN WAS TAKEN AS HOSTAGE.*SAVE ELLEN OR,I WON'T DISCUSS METAL GEAR.
Bank2:9CFA         jr      z, RescueAndShowTxt ; (!?) You can rescue him several times and upgrade your rank!
Bank2:9CFC         inc     a              ; TEXT: I'm Dr. Pettrovich...
Bank2:9CFD         call    SetAsRescued   ; Set the prisoner as rescued
Bank2:9D00
Bank2:9D00 RescueAndShowTxt:
Bank2:9D00         push   ix              ; ...
Bank2:9D02         push   af
Bank2:9D03         call    IncRescued     ; Increment the number of rescued prisoners.
Bank2:9D06         pop    af
Bank2:9D07         pop    ix
Bank2:9D09         call    SetText       ; Show the text
Bank2:9D0C
Bank2:9D0C PrisonerNextStat:
Bank2:9D0C         inc     (ix+ACTOR.Status) ; Next prisoner status
Bank2:9D0F         ret
```

# • SUBIR DE RANGO

Si intentamos rescatar al jefe.  
Pero el contador de tiempo  
que podemos repetir.

Bank2:9CF4  
Bank2:9CF4  
Bank2:9CF4  
Bank2:9CF4  
Bank2:9CF4  
Bank2:9CF4  
Bank2:9CF4  
Bank2:9CF4  
Bank2:9CF7  
Bank2:9CF8  
Bank2:9CFA  
Bank2:9CFC  
Bank2:9CFC  
Bank2:9CFD  
Bank2:9CFD  
Bank2:9D00  
Bank2:9D00  
Bank2:9D00  
Bank2:9D02  
Bank2:9D03  
Bank2:9D06  
Bank2:9D07  
Bank2:9D09  
Bank2:9D09  
Bank2:9D0C  
Bank2:9D0C  
Bank2:9D0C  
Bank2:9D0F



a ayudarnos.  
n de rango). Así

AL GEAR.

# • LÁSER

- El láser desaparece de forma rara.
- En el código se ven trazas de que podía ser de color gris o rojo.
- Quitando la llamada que fija el color, el láser desaparece correctamente.

```
Bank3:A0AD ; -----
Bank3:A0AD ;
Bank3:A0AD ; Decrease laser shot
Bank3:A0AD ; (??) This routine is weird. It seems the laser could be gray or red. Also the "erase laser" logic seems buggy
Bank3:A0AD ;
Bank3:A0AD ; -----
Bank3:A0AD LaserDecrease:
Bank3:A0AD ld a, (ix+ACTOR.LASER_WAIT) ; ...
Bank3:A0B0 rra ; (??) This field is used to set only bit0. Perhaps was LASER_CNT?
Bank3:A0B1 rra
Bank3:A0B2 ld c, 0Eh ; White color
Bank3:A0B4 jr c, LaserDecrease2
Bank3:A0B6 ld c, 8 ; Red color
Bank3:A0B8 LaserDecrease2:
Bank3:A0B8 call LaserSetSprColor ; ...
Bank3:A0B8 ; (??) The colors are ignored. Red color is always selected!
Bank3:A0B8 ; Removing this call the laser looks better
Bank3:A0BB dec (ix+ACTOR.LASER_WAIT) ; bit1:0=Red laser,1=White laser
Bank3:A0BE ret nz
Bank3:A0BF ld (ix+ACTOR.LASER_WAIT), 1 ; bit1:0=Red laser,1=White laser
Bank3:A0C3 dec (ix+ACTOR.COUNTER)
Bank3:A0C6 jr nz, EraseLaserTrace
Bank3:A0C8 |
Bank3:A0C8 jp DismissActor0
```





LIFE   
CLASS ★

 999



# • SALTO DE FE

- En el sótano del segundo edificio hay una habitación en la que había una trampa que quitaron.
- El código inicializa esa trampa para que esté abierta al entrar.

```
Bank3:A6BC ;-----  
Bank3:A6BC ;  
Bank3:A6BC ; Initialize pitfall  
Bank3:A6BC ;  
Bank3:A6BC ;-----  
Bank3:A6BC  
Bank3:A6BC InitPitfall: ; ---  
Bank3:A6BC ld hl, ColorsPitfall ; Colors used by the pitfalltiles  
Bank3:A6BF ld c, 4 ; Number of colors  
Bank3:A6C1 call SetColorsIndexes2 ; Configure colors to unpack the tiles  
Bank3:A6C4  
Bank3:A6C4 ld hl, GfxPitfall ; Pitfall compressed graphics  
Bank3:A6C7 ld de, GfxPitfallBuffer ; Buffer used to unpack the pitfall tiles  
Bank3:A6CA ld b, 0Ch  
Bank3:A6CC call SetupPitfall ; Unpack pitfall tiles and draw it fully open in URAM buffer  
Bank3:A6CF  
Bank3:A6CF ld (ix+ACTOR.COLLISION_CFG), 5 ; Enable collision with the player and set the pitfall as closed  
Bank3:A6D3  
Bank3:A6D3 ld a, (Room)  
Bank3:A6D6 cp 190 ; Isolated room in basement building 2 (CARD 1)  
Bank3:A6D8 ret nz ; (??) There is no pitfall in that room. Only a prisoner  
Bank3:A6D9  
Bank3:A6D9 ld (ix+PITFALL.Status), 2 ; Open  
Bank3:A6DD ld (ix+PITFALL.HOLE_SIZE), 3Fh ; Fully open  
Bank3:A6E1  
Bank3:A6E1 ld a, (ix+PITFALL.X) ; Pitfall center X  
Bank3:A6E4 sub 20h ; Pitfall width / 2  
Bank3:A6E6 ld (ix+PITFALL.RENDER_DX), a  
Bank3:A6E9  
Bank3:A6E9 ld a, (ix+PITFALL.Y) ; Pitfall center Y  
Bank3:A6EC sub 20h ; Pitfall height / 2  
Bank3:A6EE ld (ix+PITFALL.RENDER_DY), a  
Bank3:A6F1  
Bank3:A6F1 ld (ix+PITFALL.RENDER_SX), 40h  
Bank3:A6F5 ld (ix+PITFALL.RENDER_SY), 0A0h ; URAM page 1 coordinates of the image of a pitfall fully opened  
Bank3:A6F9 jr RenderPitfallP0 ; Draw the pitfall
```



LIFE   
CLASS ★





# • INTERRUPTOR OCULTO

En el suelo electrificado del final hay un interruptor oculto.

```
Bank5:9425          dw 7888h
Bank5:9427          db ID_LAND_MINE
Bank5:9428          dw 9898h
Bank5:942A          db ID_GAS
Bank5:942B          dw 7868h
Bank5:942D          db ID_GAS
Bank5:942E          dw 9030h
Bank5:9430          db ID_GAS
Bank5:9431          dw 68A8h
Bank5:9433
Bank5:9433 ActorsRoom115:  db 2                ; ...
Bank5:9434          db ID_CAMERA_LASER
Bank5:9435          dw 5810h
Bank5:9437          db ID_CAMERA_LASER
Bank5:9438          dw 0C010h
Bank5:943A
Bank5:943A ActorsRoom116:  db 1                ; ...
Bank5:943B          db ID_POWER_SWITCH  ; (??) It is hidden!
Bank5:943C          db 10h, 20h
Bank5:943E ActorsRoom118:  db 3                ; ...
Bank5:943F          db ID_METAL_GEAR
Bank5:9440          dw 8080h
Bank5:9442          db ID_CAMERA_LASER
Bank5:9443          dw 5818h
Bank5:9445          db ID_CAMERA_LASER
Bank5:9446          dw 0A818h
Bank5:9448
Bank5:9448 ActorsRoom119:  db 1                ; ...
Bank5:9449          db ID_BIG_BOSS
Bank5:944A          dw 3038h
Bank5:944C
```

# TRANSCEIVER



こちら JENNIFER !  
イッキ ニ ハルシカ、ナイワ!  
レ-ショ- が プレイ ダイジ ヨウブヨ。  
... OVER

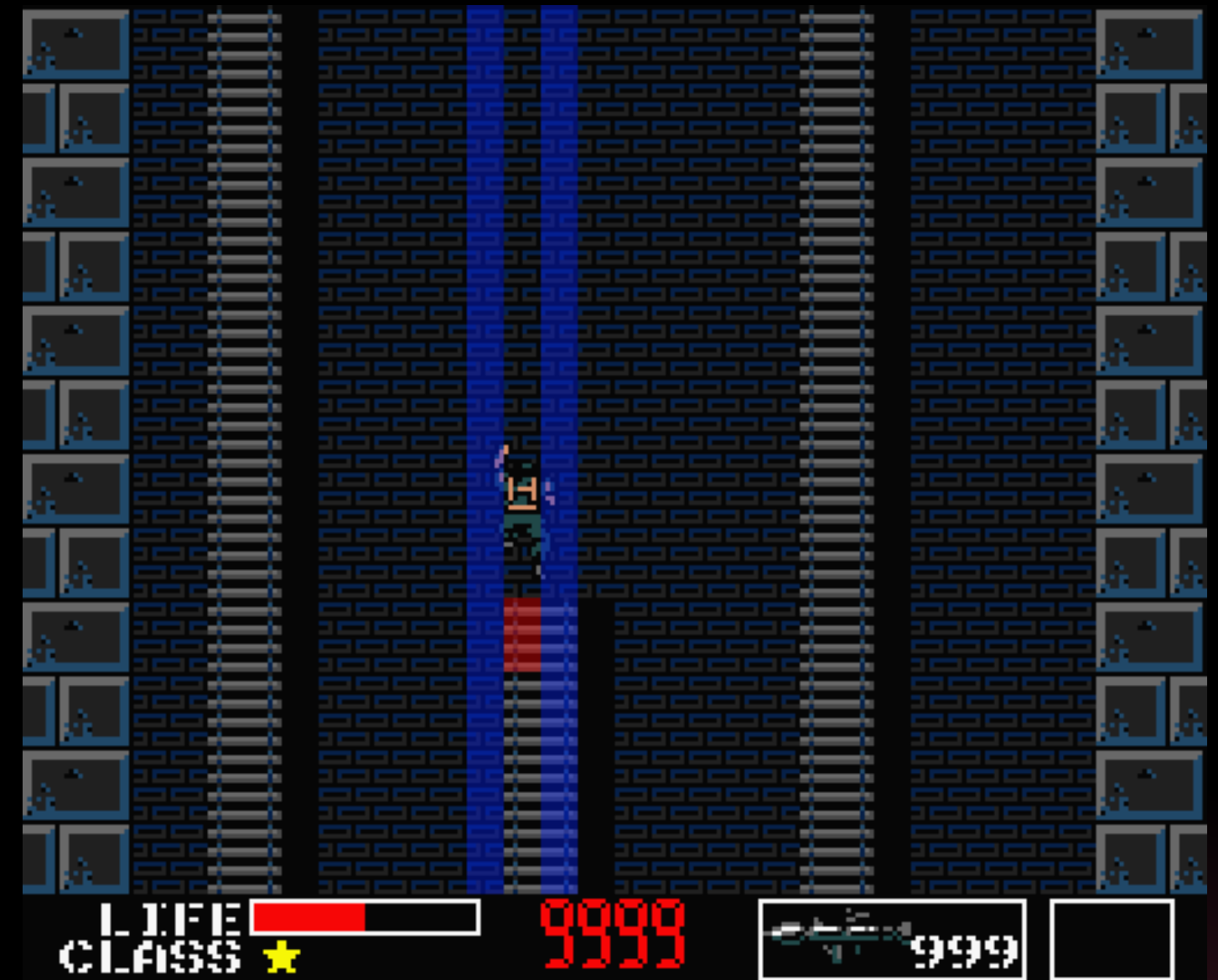
LIFE   
CLASS ★★★★★

 357

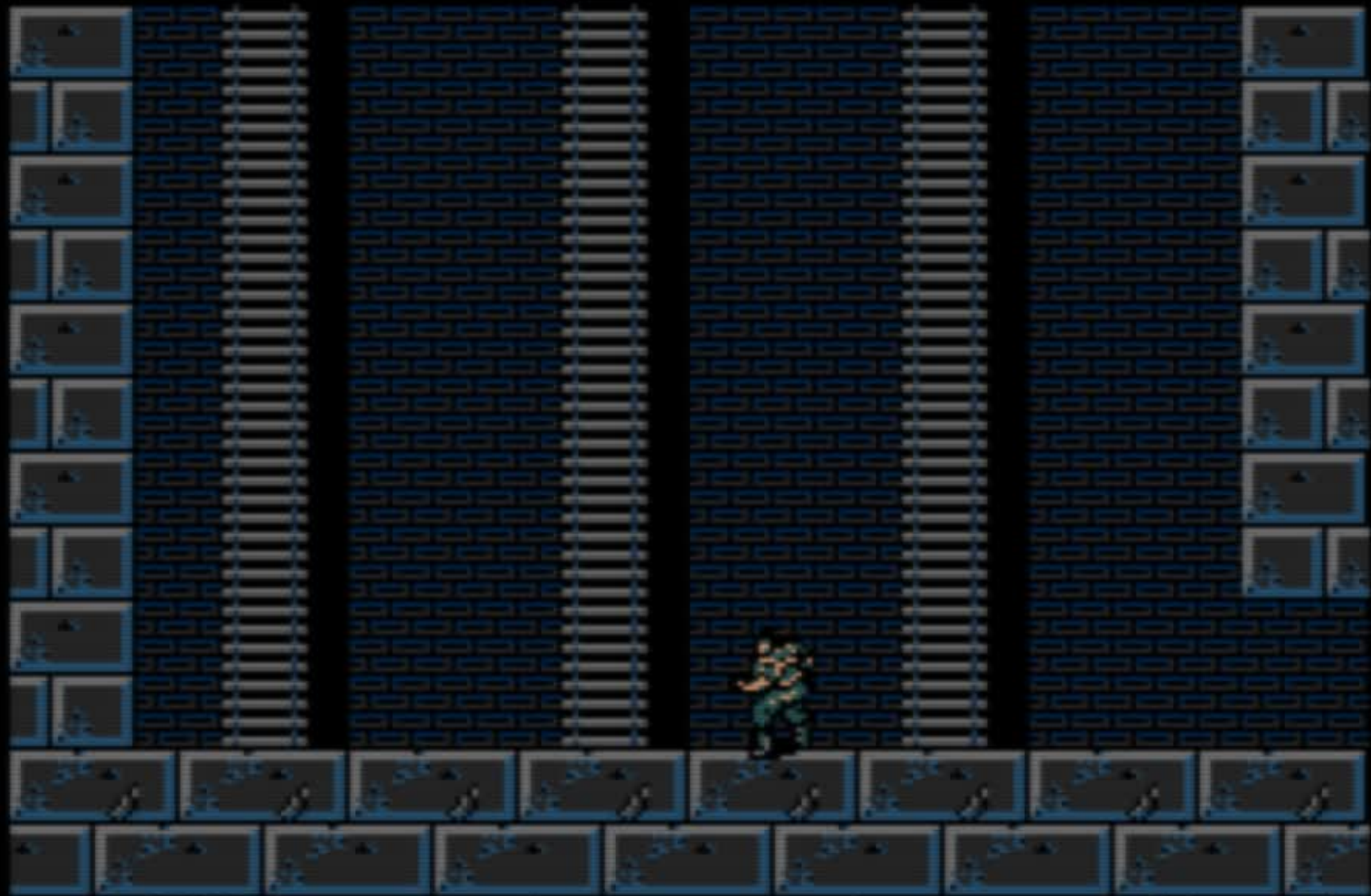
 1

# • ESCALERAS DEL FINAL

- Solo tienen tiles de colisión en la parte izquierda.
- Si no se sube centrado, se puede usar cualquier escalera para huir.







LIFE   
CLASS ★

 822



# METAL GEAR 2 - SOLID SNAKE



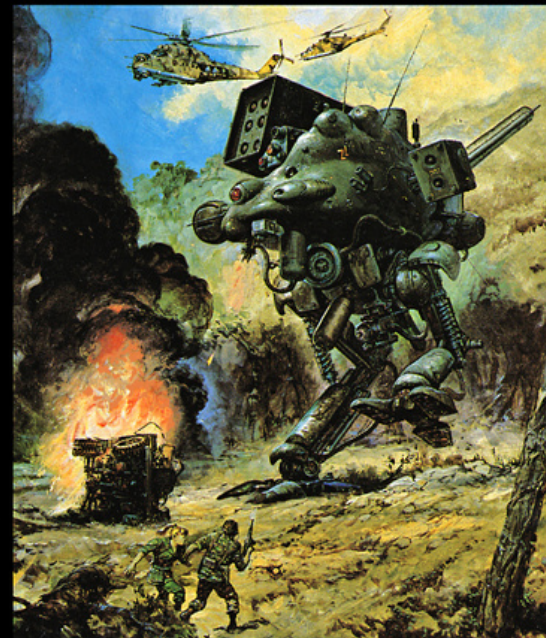
KONAMI ORIGINAL GAME SOFTWARE

RC767

# METAL GEAR 2 SOLID SNAKE™

TACTICAL ESPIONAGE GAME

© KONAMI 1990



RAM64K VRAM128KB以上  
**MSX 2 MSX 2+**  
ROM CARTRIDGE 4 MEGA 搭載

# • OBJETO 6

Al usarlo el agua te arrastra como cuando estás sumergido.

```
Bank2:9BA3          ret
Bank2:9BA4
Bank2:9BA4 ;-----
Bank2:9BA4 ;
Bank2:9BA4 ; Water drag logic
Bank2:9BA4 ;
Bank2:9BA4 ; The player is dragged by water currents.
Bank2:9BA4 ; In deep water the speed is faster
Bank2:9BA4 ; Note: using a removed item the speed is also fast on not deep water
Bank2:9BA4 ;-----
Bank2:9BA4
Bank2:9BA4 WaterDragLogic:
Bank2:9BA4          call    chkSewerCurrents__    ; ...
Bank2:9BA7          ld      a, (PlayerIsInWater)    ; 0=Not in water, 1=In water, 2=Underwater
Bank2:9BA8          or      a
Bank2:9BA9          ret      z
Bank2:9BAC          ld      de, 400h                ; Drag speed value in deep water
Bank2:9BAF          ld      a, (ItemSelected)
Bank2:9BB2          cp      ITEMS_REMOVED1        ; (??) Using this removed item the drag speed is also fast in not deep water
Bank2:9BB4          jr      z, WaterDragLogic2
Bank2:9BB6          ld      a, (PlayerIsInWater)    ; 0=Not in water, 1=In water, 2=Underwater
Bank2:9BB9          dec    a
Bank2:9BBA          jr      nz, WaterDragLogic2    ; Underwater
Bank2:9BBC          ld      d, 2                    ; Slower drag speed in not deep water
Bank2:9BBE          WaterDragLogic2:
Bank2:9BBE          ld      a, (DragDirection)    ; ...
Bank2:9BC1          or      a
Bank2:9BC2          ret      z
Bank2:9BC3          dec    a
Bank2:9BC4          call   JumpIndex
Bank2:9BC7          dw   dragSnakeLeft
Bank2:9BC9          dw   dragSnakeRight
Bank2:9BCB          dw   dragSnakeUp
Bank2:9BCD          dw   dragSnakeDown
Bank2:9BCF          dragSnakeLeft:
Bank2:9BCF          ; ...
```





ICL  
▲▲



999



# • MINAS EN EL DESIERTO NARIKO

- Hay una pantalla en la que faltan 4 minas que son sustituidas por las de otra pantalla.
- En vez de añadir las minas, sobrescribe las que había.

```
BankC:67E1 SetMinesZone_ : ; ...
BankC:67E1 ld a, (BankIn_A0)
BankC:67E4 push af
BankC:67E5 call SetBankInA0_38
BankC:67E8 call SetMinesZone
BankC:67EB pop af
BankC:67EC push af
BankC:67ED jp SetBankIn_A0
BankC:67F0 SetMinesZone: ; ...
BankC:67F0 ld hl, MineList
BankC:67F3 ld bc, 17Fh
BankC:67F6 call ClearRAM ; Clear mines
BankC:67F9
BankC:67F9 ld a, (IsolatedRoomId)
BankC:67FC and a
BankC:67FD ret nz
BankC:67FE
BankC:67FE ld a, (MapZone)
BankC:6801 cp ZONE_SWAMP
BankC:6803 jr nz, SetMinesZone2
BankC:6805
BankC:6805 ld bc, 307h ; RoomMapXY - First Nariko desert room
BankC:6808 ld hl, MinesNarikoDesert1
BankC:680B call AddMines
BankC:680E
BankC:680E ld bc, 407h ; RoomMapXY - Second Nariko desert room
BankC:6811 ld hl, MinesNarikoDesert2
BankC:6814 jr AddMines ; (!?) These mines overwrite the previous ones!
BankC:6816 SetMinesZone2: ; ...
BankC:6816 cp ZONE_METALGEAR_BASE
BankC:6818 ld bc, 208h
BankC:681B ld hl, MinesGreyFox
BankC:681E jr nz, MinesNorthArea3
```





LIFE



ICL



999





- **OBJETOS OCULTOS**



- **OBJETOS OCULTOS**





# KING KONG 2





# • OBJETOS QUE DESAPARECEN

En el altar del templo dorado hay dos objetos. Si coges solo uno desaparece el otro.

```
BB5B      ld      a, (RoomNumber)
...
BB7B
BB7B      cb      137      ; Templo dorado altar Kong
BB7D      ret      nz
BB7E
BB7E      ld      a, (NumLakunaOrb)
BB81      and      a
BB82      ret      z      ; La siguientes comprobaciones estan mal.
BB83      ; Si coges solo uno de los objetos y sales del altar
BB83      ; al volver estará el mismo objeto que cogiste y el otro habrá desaparecido.
BB83
BB83      ld      bc, 11Bh      ; Kong Roar
BB86      ld      de, 5060h
BB8C      ld      a, (NumPowerOrb) ; (!?)
BB8C      and      a
BB8D      call     z, SetItem_
BB90
BB90      ld      bc, 111h      ; Power orb
BB93      ld      de, 6060h
BB99      ld      a, (NumKongRoar) ; (!?)
BB99      and      a
BB9A      jp      z, SetItem_
BB9D      ret
```



# キングコング2

MITCHEL	1000000
D.A.M.	1000000
M.A.	1000000
M.V.E.L.	1000000
GOLD	2000000
SHOT 1	
SHOT 2	



# GOLVELLIUS

Se puede cambiar la cara del personaje del menú.

1 - Pulsar RETURN

2- Escribir uno de los siguientes nombres:

KELESIS

JEMINI

MOO

WAO

LUNARIAN

HOMME

MIYAMOTO

JANUS

YORIKI

PAC

3- Pulsar RETURN

GOTO 2





# Dragon Valley

BY PAC FUJISHIMA

COPYRIGHT©1987

COMPILE



# EXTRAÑOS BYTES AL FINAL DE LAS ROM DE KONAMI

Yie Ar Kung-Fu 1 (v2).ROM																	
Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00003F00	61	60	60	B1	E2	11	E3	61	60	60	B1	E2	11	E3	61	60	a` ` ǎ.ǎa` ` ǎ.ǎa`
00003F10	60	B1	E2	11	E3	61	60	60	B1	E2	11	E3	61	E2	61	11	ǎ.ǎa` ` ǎ.ǎaaa.
00003F20	61	E3	61	E2	61	11	61	E3	61	E2	61	11	61	E3	61	E2	aaaâa.aâââa.aâââ
00003F30	61	11	61	E3	91	E2	91	41	91	E3	91	E2	91	41	91	E3	a.aǎ ǎ Á ǎ ǎ Á ǎ
00003F40	61	E2	61	11	61	E3	61	E2	10	E3	B0	E2	11	61	E4	B1	aaa.aâââ.ǎ°â.aâǎ
00003F50	E3	B1	61	B1	21	E2	21	E3	91	E2	B1	E3	41	E2	41	E3	ǎǎǎǎǎǎǎǎǎǎǎǎǎǎ
00003F60	B1	E2	41	E4	91	E3	91	41	91	E4	B1	E3	B1	61	B1	21	ǎǎǎǎǎǎǎǎǎǎǎǎǎǎ
00003F70	E2	21	E3	91	E2	21	E3	41	E2	41	E3	B1	E2	41	E3	61	ǎ!ǎ ǎ!ǎǎǎǎǎǎǎǎǎǎ
00003F80	E2	10	E3	B0	E2	11	61	FE	FF	D7	FC	E1	61	60	60	81	ǎ.ǎ°â.apÿ>ǎǎ` `.
00003F90	61	A1	80	60	33	10	30	60	80	A4	FF	D7	FC	E1	11	10	aj€`3.0`€xy>ǎǎ..
00003FA0	10	31	11	61	30	10	E2	A3	80	A0	E1	10	30	64	FF	D8	.1.a0.ǎ&€ á.0dy0
00003FB0	FB	E1	61	40	60	D9	11	E2	B0	E1	10	41	DA	10	DB	40	úǎa@`ù.ǎ°á.Aú.ú@
00003FC0	65	FF	D8	FC	E1	11	E2	B0	E1	10	D9	E2	81	60	80	B1	ey0úǎ.ǎ°á.úǎ.`€ǎ
00003FD0	DA	80	DB	B0	E1	15	FF	D8	FC	E3	91	60	90	D9	41	10	ú€ú°á.y0úǎ` `úǎ.
00003FE0	40	60	40	DA	60	DB	80	95	FF	FF	FF	FF	FF	FF	FF	FF	@`@ú`ú€·yyyyyyyyy
00003FF0	FF	FF	FF	BA	9B	AC	85	B9	A8	80	B9	BA	81	0A	25	AA	yyy°>~..1`€¹°..%*

## YIE AR KUNG FU 1

00003FF0 FF FF FF BA 9B AC 85 B9 A8 80 B9 BA 81 0A 25 AA yyy°>~..1`€¹°..%\*

## YIE AR KUNG FU 2

00007FF0 FF 32 00 BA 9B AC 85 B9 A8 80 B9 BA 81 0C 37 AA y2.°>~..1`€¹°..7\*



# EXTRAÑOS BYTES AL FINAL DE LAS ROM DE KONAMI

**Konami**  
**イー・アル・カンフー**  
MSX

```
list
10 FOR I = &HB0 TO &HE0
20 PRINT CHR$(I);
30 NEXT
Ok
run
-アイエオカキクケコサシスセソタチツテトナニヌノハヒフ
^ホマミムメモヤユヨラリルレロフン°
Ok
█

color auto goto list run
```

**アイエオカキクケコサシスセソタチツテトナニヌノハヒフ**

Yie Ar Kung-Fu 1 (2).ROM

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00003F00	61	60	60	B1	E2	11	E3	61	60	60	61	E2	11	E3	61	60	a` ` ㄥ.ãa` ` ㄥ.ãa`
00003F10	60	B1	E2	11	E3	61	60	60	61	E2	11	E3	61	E2	11	11	ㄥ.ãa` ` ㄥ.ãaaa.
00003F20	61	E3	61	E2	61	11	61	03	61	E2	61	11	61	E3	61	E2	aaaaa.aaaaa.aaaaa
00003F30	61	11	61	E3	91	E2	91	11	91	E3	91	E2	91	41	91	E3	a.aã á Á á á Á á
00003F40	61	E3	61	11	61	E3	61	E2	10	E3	B0	E2	11	61	E4	B1	aaa.aãaa.ã°ã.aã±
00003F50	E3	B1	61	B1	2	E2	21	03	91	E2	B1	E3	41	E2	41	E3	ã±±!ã!ã!ã±±AAã
00003F60	B1	E2	41	64	91	E3	91	41	91	E4	B1	E3	B1	61	B1	21	ㄥAã á Á á ㄥ±±
00003F70	E2	21	E3	91	E2	21	E3	41	E2	41	E3	B1	E2	41	E3	61	ã!ã!ã!ãAAã ㄥAãã
00003F80	E2	10	E3	B1	E2	11	61	FE	FF	D7	FC	E1	61	00	60	81	ã.ã°ã.apyXãã` `
00003F90	61	A1	60	60	33	10	60	60	60	A	FF	D7	FC	E1	11	10	ai€ 3.0` €xyXãã..
00003FA0	10	31	1	61	30	10	E2	43	60	A	E1	10	30	64	FF	D8	.1.a0.ãã€ á.Ody0
00003FB0	FB	E1	61	40	60	D9	1	E2	60	E	11	41	DA	10	DB	40	úãã@`ù.ã°ã.Aú.ú@
00003FC0	65	FF	D8	FC	E1	1	E2	60	61	10	DA	E1	81	60	80	B1	ey0úã.ã°ã.úã. `€±
00003FD0	DA	80	DB	30	1	5	FF	D8	FC	E3	91	61	91	D9	41	10	ú€0°ã.y0úã `úã.
00003FE0	40	60	40	BA	60	68	60	65	FF	FF	FF	FF	FF	FF	FF	FF	@`@ú`ú€·yyyyyyyyy
00003FF0	FF	FF	FF	BA	9B	AC	85	B9	A8	80	B9	BA	81	0A	25	AA	yyy`>...! €'°..%ã



# EXTRAÑOS BYTES AL FINAL DE LAS ROM DE KONAMI

```
RC701 Antarctic Adventure (EN): ケッキョク ナンキョク タ"イホ"ウケン
RC703 Time Pilot[a1]: タイム ハ°イロット
RC705 Super Cobra: スーハ°ー コフ"ラ
RC706 Video Hustler: ビ"テ"オハスラー
RC714 Comic Bakery: COMIC BAKERY
RC718 Hyper Rally: ハイハ°ーラリー
RC724 Konami's Baseball: ヤキュウ
RC725 Yie Ar Kung Fu: イー・アル・カンフー
RC727 King's Valley: オウケノタニ
RC728 MopiRanger: モビ°レンシ"ャー
RC729 Pippols: ビ°ホ°ルス
RC730 Road Fighter: ロート" ファイター
RC731 Konami's Ping Pong: コナミ ノ ビ°ンホ°ン
RC732 Konami's Soccer: コナミ ノ サッカー
RC733 Hyper Sport 3: ハイハ°ー スポ°ーツ 3
RC734 Goonies: ク"ーニース"
RC735 Game Master: 10 ハ"イ タノシム カートリッシ"
RC736 Konami's Boxing: コナミ ノ ホ"クシンク"
RC737 Yie Ar Kung Fu 2: イー・アル・カンフー 2
RC739 Nightmare: マシ"ョウテ"ンセツ
RC740 Twin Bee: ツインビ"ー
RC741 Konami's Synthesizer: シンセサイサ"ー
RC742 Gradius: ク"ラテ"ィウス
RC743 Penguin Adventure: ユメタイリク アト"ハ"ンチャ"ー
RC746 QBERT: キューハ"ート
RC748 Goemon: カ"ンハ"レコ"エモン
RC749 The maze of Galious: カ"リウス ノ メイキュウ
```