

ENTERPRESS

KÉTHAVILAP AZ ENTERPRISE SZÁMÍTÓGÉPEK FELHASZNÁLÓINAK

Egy fejezet vége

1988 őszén kezdtem gondolkodni azon, hogy kellene egy Enterprise-os újságot csinálni, voltak már szerkesztői tapasztalataim. Ekkor már jól ismertem a gépet, egyedülálló képességeit, s volt sok márkatárs-barátom. Nem tetszett, hogy a szaklapok közül csak néhány szánt egy-két oldalt erre a gépre.

1989 nyarán egy véletlen folytán keveredtem Székesfehérváron az ALBA DATA Számítástechnikai Kiszövetkezethez. Itt dolgozott Annus István, akinek megemlítettem a lapalapítással kapcsolatos elképzeléseimet, meséltem neki az Enterprise-ről, a sok-sok elégedetlen felhasználóról. Aztán ennyiben maradtunk. A beszélgetésünkre kb. másfél hónappal valamért bementem az ALBA DATA-hoz, és emlékszem nagyon meglepődtem, amikor Annus úr elém lépett, és közölte, hogy megbeszélte a dolgot kollégáival, próbáljunk meg kiadni egy négy oldalas Enterprise-os mellékletet a Novopress-ben. (A Novopress egy hirdetési újság volt.) A szándékunk az volt, hogy a Novopress-en keresztül felmérjük az érdeklődést, és csak utána indítjuk el az önálló lapot.

Barátaimmal összeállítottuk a négy oldalnyi anyagot, és elkészítettük a mellékletet. A tervekben az szerepelt, hogy ősszel jelenik meg az Enterpress című hírlevél, de csak nem jött össze. Mindig kimaradt, ráadásul a Novopress szerkesztőségének az a tagja, aki a melléklettel volt megbízva, súlyos autóbalesetet szenvedett. A végén abban maradtunk, hogy a karácsonyi számban (1989. december 15.) jelenünk meg, de nem jelentünk meg. Az okokkal a mai napig sem vagyok tisztában.

Időközben az ALBA DATA-t, az Enterpress leendő kiadóját felszámolták. Ez újabb nagy pofon volt. Szerencsére Annus István nem feledkezett el a lapról. 1990 elején alakult meg a MÁTRIX Kft., s ő lett a cég ügyvezetője. A Novopress továbbra is ígérte a melléklet kiadását, de ez egészen júniusig nem történt meg, így végülis elfelejtettük őket. Ekkor döntött el: a Mátrix kiadja az Enterpress című számítástechnikai szaklapot.

Rengeteget kalkuláltunk: milyen gyakran, hány oldalon és mennyiért jelenjen meg, mennyit kell legalább eladni belőle stb. A Mátrix tulajdonosai azt állították, hogy ez a lap rengeteg munka árán a legjobb esetben is csak nullszaldós lehet, és sajnos igazuk lett. Eredetileg az Enterpress 8 oldalon és negyedévente jelent volna meg. Ezt tornáztam fel 16 oldalra és le 2 hónapra... A pénzkeret annyira szűk volt, hogy reklámra nem tudtunk költeni. Ez hiba volt, de társaimmal örültünk, hogy végre szóba áll velünk egy cég (örültek gyülekezete), és legalább annyi pénzt ad, hogy a nyomdai számlát ki tudjuk fizetni. (Később aztán reklámoztunk a szaklapokban

- Mikrovilág, Rádiótechnika, Hobbí Elektronika, Computer-world-Számítástechnika - és a Danubius Rádióban.)

Augusztusban bejegyeztettük a lapot, nyomdát kerestünk. Összeállítottuk az első számot és szeptemberben megjelent az Enterpress. Mai szemmel nézve az első példányt, nagyon kezdetlegesnek, összecsapottnak tűnik az egész. Akkor viszont nagyon büszkék voltunk rá. Aztán jött a sok-sok hálálkodó levél a felhasználóktól ("Végre valaki..."). Nagyon sok tanács érkezett a szerkesztőséghez, hogyan kellene jobban csinálni. Ezeket nagy-nagy örömmel fogadtuk, hiszen néha triviális dolgokat hagytunk ki a szerkesztésből. Fokozatosan belerázodtunk a feladatba, és tudtuk, hogy nagy-nagy hiányt pótolunk ezzel a lappal.

Az első év a tanulási volt, nagy vívódások árán kialakítottuk a lap tematikáját. Nem akartuk "témény" szakmai lappá formálni, de játékkatalógust sem akartunk szerkeszteni. Valahol a két véglet között igyekeztünk mozogni. Talán sikerült.

Akkoriban sokat álmodoztunk a színes címlapról, a 32 oldalról. Igyekeztünk feltornáznai az eladásokat, mert a kiadónak bizonyítanunk kellett. Azonban azt kellett tapasztalunk, hogy az FP tulajdonosok nagyrésze ekkorra már "lemondott" a gépről, hiszen

a 18-20 ezer felhasználó közül jó, ha 3-4 ezer tudott (tud) az Enterpress-ről. Hiába reklámoztunk, a lap nem igazán fogyott, a kiadó pedig nem jótékonyági intézmény.

Az első év végén - 1991 augusztusában - újabb embereket "fogadtunk fel" a szerkesztőségbe, és kissé módosítottuk a lap szerkezetét. És ezzel a felállással dolgoztunk mostanáig...

A Mátrix Kft. egy gazdálkodó vállalat, s nem éri meg tulajdonosainak hosszú távon sok-sok energiával egy nullszaldós kiadványt finanszírozni. Ezt tudomásul kell vennünk. En pedig, a lap felelős szerkesztőjével együtt, belefáradtam ennek a vegetálási állapotnak a fenntartásába. Ezek miatt a Mátrix felhagy a lap kiadásával, mi ketten pedig a szerkesztésével.

Ez nem jelenti azt, hogy az Enterpress megszűnik. El kell olvasni az utolsó oldal felhívását. Nagyon reméljük, hogy az új szerkesztőség és kiadó képes lesz a 300 újabb előfizető összegyűjtésére, és így nem hal meg a lap. A szerkesztőség néhány tagja - ha az olvasók is úgy akarják - folytatni fogja az írást. Sok sikert kívánunk nekik!

Végezetül szeretnék köszönetet mondani Szapper Lászlónak, Dr. Janurik Tamásnak és Bartha Istvánnak a lap tőrdeléséért; Annus Istvánnak, Juhász Istvánnak és Trautnerné Tőke Zsuzsánának támogatásukért, kitartásukért; szerkesztőtársainknak munkájukért, valamint mindazoknak, akik leveleikkel megkerestek bennünket.

Minden kedves olvasónak boldog új évet és boldogabb Enterprise-os idők eljövételét kívánja:

Hajnal Csaba

TARTALOM 92/5-6

KURZUS

Assembly — utolsó rész 2-11

PROGRAMOZÁSTECHNIKA

ISDOS betöltés 12-13

Sprite kezelés az ENTERPRISE-on 13-14

HARDVER

Memóriabővítés 320 KB-ra 15

Ha minden fólia szakad... (folyt.) 16-17

TESZT

PRINT-PACK 18-19

KÖNYVED MŰFAJ

A győztes játékprogram: CYCLOPS 20-31

MINDENFÉLE

Hirdetések, felhívások 32

ASSEMBLY

Utolsó rész

Az általam szerkesztett Assembly sorozat utolsó darabjaként két "nagy ágyút" szeretnék elsütöni, amelyeket már régóta tartogatok.

Az első a GLTX. Egy kiváló barátom írta ezt a rendkívül ügyes rendszerbővítést, amely jelentősen feljavítja a beépített szövegszerkesztőt. A programlista tanulmányozása igen tanulságos, hiszen rutinjai sok mindenről - az EXOS rendszerbővítő kialakításától kezdve a fájlkezelésen át, egészen a nyomtatóvezérlésig - szólnak.

A második POSTER MACHINE címet viselő programmal Lorigraph-fal készített LORES 2 típusú grafikákat lehet attribútum típusúvá alakítani és színeztetni. A program elsődleges célja - a sorozatra nézve - bemutatni egy gépi kódú rutinokkal teleteremtett Basic program felépítését. A Basic rész végi a teljes program vezérlését, a gépi kódú betétek pedig a sebességkritikus részeket, szolgáltatásokat valósítják meg.

Befejezésképpen minden Olvasómnak sok sikert kívánok programjaik készítéséhez. Remélem, kicsit segítetttem abban, hogy mindegyükük elindulhasson ezen az úton.

(vége)

-HCs-

GLTX a WP feljavításához

A GLTX program célja: a beépített szövegszerkesztőprogram bővítése, a szövegfeldolgozás megkönnyítése (elsősorban magnós konfigurációnál). A program lehetővé teszi:

- A teljes magyar ékezetes karakterkészlet használatát, úgy, hogy az mind az angol, mind a német billentyűzetű gépeken egyaránt használható legyen.

- Vezérlőkarakterek elhelyezését a szövegben, amit nem a program értelmez, hanem közvetlenül a printer és így lehetővé teszi egy EPSON-kompatibilis nyomtató valamennyi lehetőségének kihasználását.

- Továbbá egy érdekes szolgáltatást is nyújt a GLTX! Ez a TXRAM funkció. Ez tulajdonképpen nem más, mint egy háttér EDITOR, amelybe elküldhetjük a szövegszerkesztőben lévő szöveget, majd bármikor - a kurzor pozíciójától kezdődően - visszatölthetjük. Lehetőség van a TXRAM tartalmának háttértárolón történő tárolására és visszatöltésére is.

A GLTX-nek van egy saját help-táblája, amit bármikor megtekinthetünk. Ezen a GLTX parancsai vannak. A help-tábla megtekintése WP-ből:

Üssük le az F8 billentyűt, majd gépeljük be: HELP GLTX.

A tábla ekkor megjelenik és az ENTER billentyű leütésével térhetünk vissza a szövegbe. Ennek a help-táblának az az előnye, hogy a HELP GLTX parancsra nem az aktuális szerkesztőcsatornára írja a help-et, hanem megjelenít egy táblát, és így nem rontja el a WP-ben szerkesztett szöveget.

A GLTX betöltése

Betöltés WP-ből: F1

Betöltés BASIC-ből: F1 vagy LOAD

A program betöltése után látszólag nem történik semmi, mert a GLTX egy rendszerbővítés. A program jelenlétét ellenőrizhetjük, ha BASIC-ben kiadjuk a :HELP parancsot.

A magyar ékezetes karakterkészlet

A magyar ékezetes karakterkészletet a :GLH parancsral aktivizálhatjuk. A karakterkészlet alapállapotba állítása a :GLTX parancsral történik.

A magyar betűk definiálásán kívül a program 5 funkcióbillentyűt is definiál. Ezek a következők:

- SHIFT+F1 Szövegrész törlése kurzor alatt behúzással
- SHIFT+F2 Szövegrész törlése kurzor felett behúzással
- SHIFT+F3 Szövegrész törlése aluról felfelé
- SHIFT+F4 Szövegrész törlése felülről lefelé
- SHIFT+F5 Sorok bekezdésbe szerkesztése

Szövegek nyomtatása

a szövegszerkesztő tartalmának kinyomtatását a :GLP parancsral indíthatjuk. Ekkor a program megkérdezi az egység és az állomány nevét, majd megkezdji a szöveg elküldését. Az egység és filenév megadására azért van szükség, mert így nem csak a nyomtatóra, hanem pl. kazettára vagy lemezre is vihetjük a szöveget vezérlőkarakterekkel együtt.

A GLP a következőket figyeli a szöveg nyomtatása közben:

- Ha talál egy ASCII 155 kódú karaktert, akkor e karakter helyett ASCII 27-et küld. Így egy escape szekvenciát küldhetünk a printernek, amellyel bizonyos nyomtatási képek állíthatók be. Ezeket az adott EPSON kompatibilis nyomtató kézikönyvéből nézhetjük ki.

- Ha a program talál egy ASCII 156) /ALT& a német gépeken/ karaktert,

- felfüggeszti a nyomtatást és vár mindaddig, míg le nem ütjük az ENTER

- billentyűt. Ez a PAUSE funkció. Ilyenkor a keret színe kék-ről pirosra

- változik, jelezvén, hogy billentyűlcütere vár.

- Ha a program kettő darab CHR\$(156) kódú karaktert talál, akkor végrehajt

- egy lapdobást. (FF : Form Feed)

- A nyomtatás az ESC billentyűvel is megállítható.

(Ezeket a kódokat természetesen egyszerű billentyűlenyomással tesszük a WP-be. Német gépeken az ASCII 156-hoz például az [ALT] billentyűket. A szerk.)

A TXRAM

A TXRAM tulajdonképpen egy háttér EDITOR. Ebbe elmenthetjük a szövegszerkesztő tartalmát, és bármikor visszatölthetjük a kurzor pozíciójától kezdődően. A TXRAM-ban lévő szöveget eltávolíthatjuk háttértárolón, vagy kinyomtatathatjuk.

A TXRAM parancsai:

- TXDEF - létrehozza a TXRAM-ot. (Ha a rendszer nem tud elegendő mennyiségű RAM-ot lefoglalni, hibaüzenetet kapunk.)

- TXCLR - Törli a TXRAM-ban lévő szöveget. (A TXRAM-ot nem szünteti meg.)

- TXSAVE - A szövegszerkesztőben lévő szöveget átmásolja a TXRAM-ba.

- TXLOAD - A TXRAM-ban lévő szöveget másolja át a szövegszerkesztőbe.

- SAVETX - A TXRAM-ban lévő szöveget menti háttértárolóra editor formátumban.

- Az egység és állománynevet kéri. Alapértelmezés: TAPE:1. LOADTX - Háttértárról a TXRAM-ba tölt egy file-t. ASCII és editor formátumú állományok egyaránt betölthetők. Bekéri az egység és állománynevet.

- TXPRN - A TXRAM-ban lévő szöveget küldi ki a megadott perifériára ASCII formában. Alapértelmezés: PRINT:R.

- Hogy mire jó a TXRAM? Például:

- Egy ismétlődő szövegrészt eltávolítunk a TXRAM-ban, és amikor csak kell előhívjuk. Így sok felesleges gépeléstől mentesülünk.

- Ha váltani akarunk 40 vagy 80 karakteres üzemmód között, a WP tartalma elvész, ezért itt is használhatjuk a TXRAM-ot, mert ez megőrzi tartalmát.

- Használható még editor formátumú állományok összefűzésére is. Amikor a Wp betölt egy szöveget, törli a már meglévőket. Így nem tudunk szövegeket összemásolni. De, ha a hozzáfűzendő szöveget a TXRAM-ba töltjük és ebből másoljuk át a szövegszerkesztőbe, akkor megoldható ez a probléma is.

- Ráadásul a GLTX bármilyen alkalmazói programból meghívható!

Tájékoztatóként közlöm, hogy a TXRAM a 254-es csatornán van megnyitva és tartozik hozzá egy x=40 y=3 méretű 80 karakteres vidcolap a 109-es csatornán.

- Göbölös László -

;GLTX (c)1988 by Ladislav

```

gltx  ld a,c
      dec a
      dec a
      jr z,indul0
      dec a
      ret nz
help0 or b
      jr z,nagyh0
      call contro0
      ret nz
      ld a,0ffh
      push bc
      push de
      call hpscr
      ld a,113d
      ld bc,vege2-szovegg
      ld de,szovegg
      exos 8
      ld a,113d
      ld b,1
      ld c,1d
      ld d,20d
      ld e,4d
      exos 11d ;display video
cikl  ld a,07d
      out (0b5h),a
      in a,(0b5h)
      sub 191d
      jr z,excikl
      jr cikl
excikl ld a,113d
      exos 3
      pop de
      pop bc
      xor a
      ld c,a
      ret
nagyh0 ld a,0ffh
      push bc
      push de
      ld bc,vege1-szovegg
      ld de,szovegg
      exos 8
      pop de
      pop bc
      xor a
      ret
indul0 call contro0
      jp nz,txsave
      ld (regist0),sp
      call fprg0
      ld sp,(regist0)
      xor a
      ld c,a
      ret
contro0 push bc
      push de
      push hl
      ld hl,szoveg0
ellen0  inc de
      inc hl
      ld a,(de)
      cp (hl)
      jr nz,nemjo0
      djnz ellen0
nemjo0 pop hl
      pop de
      pop bc
      ret
szoveg0 defb 4
      defm "GLTX"
regist0 defw 0
fprg0  ld b,0
      ld c,29
      exos 16
      ld a,d
      push af
      ld b,4
      exos 11
      ld c,0
      ld sp,(regist0)

```

```

      ret
;
txsave ld a,c
      dec a
      dec a
      jr z,indul
      dec a
      ret nz
indul  call contro1
      jr nz,txload
      ld (regist),sp
      call fprg
      ld sp,(regist)
      xor a
      ld c,a
      ret
control push bc
      push de
      push hl
      ld hl,szoveg
ellen  inc de
      inc hl
      ld a,(de)
      cp (hl)
      jr nz,nemjo
      djnz ellen
nemjo  pop hl
      pop de
      pop bc
      ret
szoveg defb 6
      defm "TXSAVE"
regist defw 0
fprg  ld b,1h
      ld c,20h
      ld d,0c0h
      exos 10h
manci ld a,3d
      out (0b5h),a
      in a,(0b5h)
      sub 127d
      jr z,vissza
      ld a,255d
      exos 5d
      sub 0e4h
      jr z,vissza
      ld a,254d
      exos 7d
      jr manci
vissza ld sp,(regist)
      ld c,0
      ret
txload ld a,c
      dec a
      dec a
      jr z,indul2
      dec a
      ret nz
indul2 call contro2
      jr nz,txclr
      ld (regist2),sp
      call fprg2
      ld sp,(regist2)
      xor a
      ld c,a
      ret
contro2 push bc
      push de
      push hl
      ld hl,szoveg2
ellen2  inc de
      inc hl
      ld a,(de)
      cp (hl)
      jr nz,nemjo2
      djnz ellen2
nemjo2 pop hl
      pop de
      pop bc
      ret
szoveg2 defb 6
      defm "TXLOAD"

```

```

regist2 defw 0
fprg2   ld b,1h
        ld c,20h
        ld d,0c0h
        exos 10h
pisti   ld a,3d
        out (0b5h),a
        in a,(0b5h)
        sub 127d
        jr z,vissza2
        ld a,254d
        exos 5d
        sub 0e4h
        jr z,vissza2
        ld a,255d
        exos 7d
        jr pisti
vissza2 ld sp,(regist2)
        ld c,0
        ret
txclr   ld a,c
        dec a
        dec a
        jr z,indul3
        dec a
        ret nz
indul3  call contro3
        jr nz,txdef
        ld (regist3),sp
        call fprg3
        ld sp,(regist3)
        xor a
        ld c,a
        ret
;
contro3 push bc
        push de
        push hl
        ld hl,szoveg3
ellen3  inc de
        inc hl
        ld a,(de)
        cp (hl)
        jr nz,nemjo3
        djnz ellen3
nemjo3  pop hl
        pop de
        pop bc
        ret
szoveg3 defb 5
        defm "TXCLR"
regist3 defw 0
fprg3   ld a,254d
        ld b,1ah
        exos 7d
        ld sp,(regist3)
        ld c,0
        ret
txdef   ld a,c
        dec a
        dec a
        jr z,indul4
        dec a
        ret nz
indul4  call contro4
        jp nz,glp
        ld (regist4),sp
        call fprg4
        ld sp,(regist4)
        xor a
        ld c,a
        ret
contro4 push bc
        push de
        push hl
        ld hl,szoveg4
ellen4  inc de
        inc hl
        ld a,(de)
        cp (hl)
        jr nz,nemjo4
        djnz ellen4
nemjo4  pop hl
        pop de
        pop bc
        ret
szoveg4 defb 5
        defm "TXDEF"
regist4 defw 0
fprg4   ld b,1
        ld c,22d
        ld d,2d
        exos 16d ;video mode
        ld b,1
        ld c,24d
        ld d,40d
        exos 16d ;video x
        ld b,1
        ld c,25d
        ld d,3d
        exos 16d ;video y
        ld b,1
        ld c,23d
        ld d,0
        exos 16d ;video color
        ld a,109d
        ld de,vid
        exos 1
editor  ld b,1
        ld c,31d
        ld d,250d
        exos 16d
        ld b,1
        ld c,29d
        ld d,109
        exos 16d
        ld a,254d
        ld de,edi
        exos 1d
        ld sp,(regist4)
        ld c,0
        ret
edi     defb 7
        defm "editor:"
vid     defb 6
        defm "video:"
glp     ld a,c
        dec a
        dec a
        jr z,indul5
        dec a
        ret nz
indul5  call contro5
        jp nz,glh
        ld (regist5),sp
        call fprg5
        ld sp,(regist5)
        xor a
        ld c,a
        ret
contro5 push bc
        push de
        push hl
        ld hl,szoveg5
ellen5  inc de
        inc hl
        ld a,(de)
        cp (hl)
        jr nz,nemjo5
        djnz ellen5
nemjo5  pop hl
        pop de
        pop bc
        ret
szoveg5 defb 3
        defm "GLP"
regist5 defw 0
fprg5   jp prnopen
kezdel  ld b,1
        ld c,01bh
        ld d,24h
        exos 10h
        ld b,1h
        ld c,20h

```

```

ld d,0c0h
exos 10h
olvas ld a,3d
out (0b5h),a
in a,(0b5h)
sub 127d
jr z,bef
ld a,0ffh
exos 5
sub 0e4h
jr z,bef
ld a,b
sub 155d
jr z,cape
ld a,b
sub 09ch
jr z,allj
ld a,160d
exos 7
jr olvas
cape ld a,160d
ld b,01bh
exos 7
jr olvas
allj ld a,0FFh
exos 5
ld a,b
sub 09ch
jr z,newlap
wait ld b,01h
ld c,01bh
ld d,049h
exos 10h
wait2 ld a,07d
out (0b5h),a
in a,(0b5h)
sub 191d
jr z,olvasd
jr wait2
olvasd ld b,01
ld c,01bh
ld d,024h
exos 10h
jr olvas
bef jp prncIs
viszsa ld b,1
ld c,01bh
ld d,0
exos 10h
ld sp,(regist5)
ld c,0
ret
newlap ld a,160d
ld b,12d
exos 7d
jp olvas
glh ld a,c
dec a
dec a
jr z,indul8
dec a
ret nz
indul8 call contro8
jp nz,savetx
ld (regist8),sp
call fprg8
ld sp,(regist8)
xor a
ld c,a
ret
contro8 push bc
push de
push hl
ld hl,szoveg8
ellen8 inc de
inc hl
ld a,(de)
cp (hl)
jr nz,nemjo8
djnz ellen8
nemjo8 pop hl
pop de
pop bc
ret
szoveg8 defb 3
defm "GLH"
regist8 defw 0
fprg8 ld b,0
ld c,30d
exos 16d ;ask editor key
ld a,d
ld b,8d
ld c,8d
ld de,fkey1
exos 11d
ld b,0
ld c,30d
exos 16d ;ask editor key
ld a,d
ld b,8d
ld c,9d
ld de,fkey2
exos 11d
ld b,0
ld c,30d
exos 16d ;ask editor key
ld a,d
ld b,8d
ld c,10d
ld de,fkey3
exos 11d
ld b,0
ld c,30d
exos 16d ;ask editor key
ld a,d
ld b,8d
ld c,11d
ld de,fkey4
exos 11d
ld b,0
ld c,30d
exos 16d ;ask editor key
ld a,d
ld b,8d
ld c,12d
ld de,fkey5
exos 11d
ld b,0
ld c,29
exos 16;video csat.lekerdezese
ld a,d
push af
ld b,4
pop af
ld bc,20*12
ld de,kezdet
exos 8 ;blokk irasa
ld c,0
ld sp,(regist8)
ret
fkey1 defb 2d,160d,161d
fkey2 defb 2d,164d,165d
fkey3 defb 2d,176d,161d
fkey4 defb 2d,180d,161d
fkey5 defb 4d,164d,168d,180d,185d
kezdet defb 1bh,'K',145,12,24,60,102,126,102,102,0,0
defb 1bh,'K',151,12,24,126,96,120,96,126,0,0
defb 1bh,'K',133,12,24,60,24,24,24,60,0,0
defb 1bh,'K',146,12,24,60,102,102,102,102,60,0
defb 1bh,'K',148,54,54,60,102,102,102,102,60,0
defb 1bh,'K',154,12,24,102,102,102,102,60,0
defb 1bh,'K',149,27,54,0,102,102,102,102,60,0
defb 1bh,'K',129,12,24,60,6,62,102,62,0,0
defb 1bh,'K',147,12,24,60,102,126,96,60,0,0
defb 1bh,'K',132,12,24,56,24,24,24,60,0,0
defb 1bh,'K',134,12,24,60,102,102,102,60,0,0
defb 1bh,'K',135,27,54,60,102,102,102,60,0,0
defb 1bh,'K',136,12,24,102,102,102,102,60,0,0
defb 1bh,'K',138,27,54,0,102,102,102,60,0,0
defb 1bh,'K',155,16,24,28,30,31,30,28,24,16
defb 1bh,'K',156,60,126,255,255,255,255,126,60
defb 1bh,'K',139,102,0,60,102,102,102,60,0,0
defb 1bh,'K',140,102,0,102,102,102,102,60,0,0
defb 1bh,'K',137,102,60,102,102,102,102,60,0,0
defb 1bh,'K',143,102,66,102,102,102,102,60,0,0

```

```

szovegg defb 4
         defm "GLTX version 1.10 -(c) Ladislás-"
         defb 13,10
vege1   ;
         defb 1bh,'c',73,219,88,88,0,0,0,0
         defm "      Written by Ladislás (c)1988"
         defb 13,10
         defm "  GLH  Hungary Mode and new FKEYS"
         defb 13,10
         defm "  GLTX clear font"
         defb 13,10
         defm "  GLP  printer program : "
         defb 13,10
         defm "      ALT+ + [ = ESCape to printer"
         defb 13,10
         defm "      ALT+ > \ = PAUSE  printing"
         defb 13,10
         defm "      ALT+ >> \\ = FF    Form Feed"
         defb 13,10
         defm "      ENTER  = Continue printing"
         defb 13,10
         defm "      ESC    = Break   printing"
         defb 13,10
         defm "  TXDEF make the TXRAM"
         defb 13,10
         defm "  TXCLR clear the TXRAM"
         defb 13,10
         defm "  TXSAVE save text into TXRAM"
         defb 13,10
         defm "  TXLOAD load text from TXRAM"
         defb 13,10
         defm "  SAVETX save the TXRAM"
         defb 13,10
         defm "  LOADTX load the TXRAM"
         defb 13,10
         defm "  TXPRN print the TXRAM"
         defb 13,10
         defm "F9-F12 delete lines . F13 arrange"
         defb 13,10
         defb 13,10
         defm "      Press ENTER !"
vege2   ;
         defm "Ladislás Software"
         ;
first2  defb 27,44,145,133
         defb 27,44,151,166
         defb 27,44,133,173
         defb 27,44,146,168
         defb 27,44,148,167
         defb 27,44,154,169
         defb 27,44,149,170
         defb 27,44,129,160
         defb 27,44,147,136
         defb 27,44,132,161
         defb 27,44,134,162
         defb 27,44,136,163
         defb 27,44,138,165
         defb 27,44,135,164
         defb 27,44,137,131
         defb 27,44,143,132
         defb 27,44,139,147
         defb 27,44,140,148
last2   ;
prnopen ld de,nyomatat
         ld c,0
         exos 19d
         call input
         ld a,160d
         ld de,flen
         exos 2d
         ld a,160d
         ld bc,last2-first2
         ld de,first2
         exos 8
         jp kezdel
prncis ld de,magno
         ld c,1d
         exos 19d
         ld a,160d
         exos 3
         jp visszae
nyomatat defb 7
         defm "PRINTER:"
magno   defb 4
         defm "TAPE:"
         defm "
puffer  defm "
savetx  ld a,c
         dec a
         dec a
         jr z,indul9
         dec a
         ret nz
indul9  call contro9
         jp nz,loadtx
         ld (regist9),sp
         call fprg9
         ld sp,(regist9)
         xor a
         ld c,a
         ret
contro9 push bc
         push de
         push hl
         ld hl,szovegg9
ellen9  inc de
         inc hl
         ld a,(de)
         cp (hl)
         jr nz,nemjo9
         djnz ellen9
nemjo9  pop hl
         pop de
         pop bc
         ret
szovegg9 defb 6
         defm "SAVETX"
vege9   ;
regist9 defw 0
fprg9   call input
         ld a,106d
         ld de,flen
         exos 2d
         ld a,254d
         ld b,26d
         ld c,106d
         exos 11d
         ld a,106d
         exos 3d
         ld sp,(regist9)
         ld c,0
         ret
loadtx  ld a,c
         dec a
         dec a
         jr z,indulq
         dec a
         ret nz
indulq  call controq
         jp nz,txprn
         ld (registq),sp
         call fprgq
         ld sp,(registq)
         xor a
         ld c,a
         ret
controq push bc
         push de
         push hl
         ld hl,szovegg
ellenq  inc de
         inc hl
         ld a,(de)
         cp (hl)
         jr nz,nemjoq
         djnz ellenq
nemjoq  pop hl
         pop de
         pop bc
         ret
szovegg9 defb 6
         defm "LOADTX"
vegeq   ;
registq defw 0
fprgq   call input
         ld a,106d
tolt    ld a,106d

```

```

ld de,flen
exos 1d
ld de,header
ld a,106d
exos 29d
ld a,254d
ld b,25d
ld c,106d
exos 11d
adc a,1d
sub 1d
jr z,lezar
csital ld a,106d
exos 5d
sub 0e4h
jr z,lezar
ld a,254d
exos 7d
jr csital
lezar ld a,106d
exos 3d
ld sp,(registq)
ld c,0
ret
flen defs 1
fname defn "....."
input ld b,1
ld c,22d
ld d,0
exos 16d ;video mode
ld b,1
ld c,24d
ld d,40d
exos 16d ;video x
ld b,1
ld c,25d
ld d,3d
exos 16d ;video y
ld b,1
ld c,23d
ld d,0
exos 16d ;video color
ld a,111d
ld de,vid
exos 1d ;video open
ld b,0
ld c,30d
exos 16d ;ask editor key
ld b,1
ld c,30d
exos 16d ;set editor key
ld c,29d
ld d,111d
exos 16d ;editor video
ld b,1
ld c,31d
ld d,1
exos 16d ;editor buffer
ld a,112d
ld de,edi
exos 1d ;open editor
ld a,111d
ld b,1d
ld c,1d
ld d,3d
ld e,1d
exos 11d ;display video
ld bc,pal2-pal1
ld de,pal1
ld a,111d
exos 8 ;set palette
ld bc,tyfl2-tyfl1
ld de,tyfl1
ld a,112d
exos 8
ld b,1d
ld c,32d
ld d,00011000b
exos 16d ;flags editor
ld hl,fname
read ld a,112d
exos 5d ;input from editor
ld a,b
sub 13d
jr z,endingp
ld (hl),b
inc hl
jr read
endingp inc hl
ld (hl),13d
ld hl,fname
ld d,0
pipi ld a,(hl)
sub 13d
jr z,kate
inc d
inc hl
jr pipi
ld hl,flen
dec d
ld (hl),d
ld a,111d
exos 3d
ld a,112d
exos 3d
ret
tyfl1 defm "Type: DEVICE:FILENAME and press ENTER!"
defb 13,10
;
tyfl2 defm "....."
pal1 defb 1bh,'C',73d,219d,146d,146d,0,0,0,0
pal2 ;
hpscr ;
ld b,1
ld c,22d
ld d,0
exos 16d ;video mode
ld b,1
ld c,24d
ld d,40d
exos 16d ;video x
ld b,1
ld c,25d
ld d,20d
exos 16d ;video y
ld b,1
ld c,23d
ld d,0
exos 16d ;video color
ld a,113d
ld de,vid
exos 1d ;open video
ret
txprn ld a,c
dec a
dec a
jr z,indulw
dec a
ret nz
indulu call controw
ret nz
ld (registw),sp
call fprgw
ld sp,(registu)
xor a
ld c,a
ret
controw push bc
push de
push hl
ld hl,szovegw
ellenw inc de
inc hl
ld a,(de)
cp (hl)
jr nz,nemjow
djnz ellenw
nemjow pop hl
pop de
pop bc
ret
szovegw defb 5
defm "TXPRN"
vegew ;
registw defw 0
fprgw ld de,nyontat
ld c,0
exos 19d
call input
ld a,160d
ld de,flen
exos 2d
ld b,1h
ld c,20h
ld d,0c0h
exos 10h
olvasu ld a,3d
out (0b5h),a
in a,(0b5h)
sub 127d
jr z,befw
ld a,0feh
exos 5
sub 0e4h
jr z,befw
ld a,160d
exos 7
jr olvasu
befw ld de,magno
ld c,1d
exos 19d
ld a,160d
exos 3
ld sp,(registw)
ld c,0
ret

```

A POSTER MACHINE programlistája

```

100 PROGRAM "POSTER"
110 ALLOCATE 1100
120 SET INTERRUPT STOP OFF
130 WHEN EXCEPTION USE LETSGO
140 CLEAR FONT
150 SET FKEY 1 "LORI3"
160 SET FKEY 2 "TAPE:"
170 SET FKEY 3 "DISK:"
180 CLEAR SCREEN
190 SET 27,0
200 TEXT 40
210 SET 26,1
220 SET 6,1
230 SET #102:PALETTE 0,CYAN,0,YELLOW
240 CODE RTN=HEX$( "0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0" )
250 CODE BOX1=HEX$( "E5,E5,DD,E1,DD,46,4,16,0,DD,5E,1,21,0,0,19,10,FD,DD,5E,
0,19,DD,5E,E,DD,56,F,19,DD,75,7,DD,74,8" )
260 CODE =HEX$( "DD,e1,DD,6E,7,DD,66,8,11,0,CO,DD,46,3,DD,4E,2,E5,7E,12,DD,
7E,B,77,23,13,D,20,F5,E1,D5,16,0,DD,5E,4,19,D1,10,E6,C9" )
270 CODE PAP=HEX$( "3E,1,D3,B5,DB,B5,FE,BF,28,D,FE,DF,28,14,FE,F7,28,2F,FE,
EF,28,36,CO" )
280 CODE =HEX$( "DD,7E,5,3D,E6,F,DD,77,5,18,9,DD,7E,5,3C,E6,F,DD,77,05,CB,
27,CB,27,CB,27,CB,27,26,0,6F,6,0,DD,4E,6,9,7D,DD,77,B,C9" )
290 CODE =HEX$( "DD,7E,6,3D,E6,OF,DD,77,6,18,9,DD,7E,6,3C,E6,F,DD,77,6,26,
0,6F,6,0,DD,4E,5,CB,21,CB,21,CB,21,CB,21,09,7D,DD,77,B,C9" )
300 CODE KURSIV=HEX$( "21,80,b4,1,40,2,cb,3e,23,b,78,b1,20,f8,c9" )
310 CODE BITMAP=HEX$( "3E,65,6,3,F7,B,C5,E1,C9" )
320 CODE ATRMAP=HEX$( "3E,65,6,3,F7,B,D5,E1,C9" )
330 CODE BITLD=HEX$( "F3,3E,FE,D3,B1,FB,3E,11,1,0,0,11,0,0,F7,6,3E,11,F7,3,
F3,3E,F9,D3,B1,FB,C9" )
340 CODE BILL=HEX$( "E5,3E,69,6,9,E,0,F7,B,DD,E1,79,FE,1,28,2D,FE,2,28,39,
FE,4,28,3F,FE,8,28,4A,FE,9,28,50,FE,5,28,53,FE,6,28,56,FE,A,28,59" )
350 CODE =HEX$( "FE,11,28,5C,FE,12,28,68,FE,14,28,6E,FE,18,28,79,C9" )
360 CODE JOBBRA=HEX$( "DD,7E,0,DD,86,2,DD,46,4,0,B8,C8,DD,34,0,C9" )
370 CODE BALRA=HEX$( "DD,7E,0,FE,0,C8,DD,35,0,C9" )
380 CODE LE=HEX$( "DD,7E,1,DD,86,3,DD,46,D,B8,C8,DD,34,1,C9" )
390 CODE FEL=HEX$( "DD,7E,1,FE,0,C8,DD,35,1,C9" )
400 CODE JF=HEX$( "CD" )&WORDS$( JOBBRA )&HEX$( "CD" )&WORDS$( FEL )&HEX$( "C9" )
410 CODE JL=HEX$( "CD" )&WORDS$( JOBBRA )&HEX$( "CD" )&WORDS$( LE )&HEX$( "C9" )
420 CODE BL=HEX$( "CD" )&WORDS$( BALRA )&HEX$( "CD" )&WORDS$( LE )&HEX$( "C9" )
430 CODE BF=HEX$( "CD" )&WORDS$( BALRA )&HEX$( "CD" )&WORDS$( FEL )&HEX$( "C9" )
440 CODE =HEX$( "DD,7E,0,DD,86,2,DD,46,4,0,B8,C8,DD,34,2,C9" )
450 CODE =HEX$( "DD,7E,2,FE,1,C8,DD,35,2,C9" )
460 CODE =HEX$( "DD,7E,1,DD,86,3,DD,46,D,B8,C8,DD,34,3,C9" )
470 CODE =HEX$( "DD,7E,3,FE,1,C8,DD,35,3,C9" )
480 CODE BOX2=HEX$( "E5,DD,e1,11,0,CO,DD,6E,7,DD,66,8,DD,46,3,DD,4E,2,E5,
1A,77,23,13,D,20,F9,E1,D5,16,0,DD,5E,4,19,D1,10,EA,C9" )
490 CODE BOX3=HEX$( "E5,DD,e1,11,0,CO,DD,6E,7,DD,66,8,DD,46,3,DD,4E,2,E5,
1A,D5,CB,BF,CB,B7,CB,AF,CB,A7,DD,5E,5,CB,3,CB,3,CB,3,83,D1,77,23,
13,D,20,E3,E1,D5,16,0,DD,5E,4,19,D1,10,D4,C9" )
500 CODE BOX4=HEX$( "E5,DD,E1,11,0,CO,DD,6E,7,DD,66,8,DD,46,3,DD,4E,2,E5,
1A,D5,CB,9F,CB,97,CB,8F,CB,87,DD,5E,6,83,D1,77,23,13,D,20,eb,E1,D5,
16,0,DD,5E,4,19,D1,10,dc,C9" )
510 CODE IO=HEX$( "7E,D3,B5,DB,B5,23,C9" )
520 CODE SEQ=HEX$( "1B,43,0,0,0,0,0,0,0,0,0,0" )
530 CODE TAB=HEX$( "3,FD,2,FD,3,BF,2,BF,3,DF,2,DF,3,F7,2,F7,3,EF,2,EF,3,
FB,2,FB,3,FE,2,FE,5,FE,9,FE,0,0,0" )
540 CODE COLORS=HEX$( "FD,21" )&WORDS$( SEQ+2 )&HEX$( "21" )&WORDS$( TAB )&HEX$( "6,
8,CD" )&WORDS$( IO )&HEX$( "BE,28,26,23,CD" )&WORDS$( IO )&HEX$( "BE,28,24,FD,
23,23,10,EE" )
550 CODE =HEX$( "3E,5,D3,B5,DB,B5,FE,FB,CO,6,0,E,1C,F7,10,7A,C6,8,57,6,1,
E,1C,F7,10,C9,FD,34,0,18,3,FD,35,0,3E,65,1,A,
0,11" )&WORDS$( SEQ )&HEX$( "F7,8,C9" )
560 CODE LEFT=HEX$( "21,0,0,6,0,37,3f,16,0,cb,16,2b,15,7a,20,f9,38,3,10,
f1,c9,3f,c5,1,0,0,9,cb,c6,ed,42,c1,18,f0" )
570 CODE RIGH=HEX$( "21,0,0,6,0,37,3f,16,0,cb,1E,23,15,7a,20,f9,38,3,10,
f1,c9,3f,c5,1,0,0,ED,42,cb,FE,9,c1,18,f0" )
580 CODE SCL=HEX$( "3E,0,D3,B5,DB,B5,FE,DF,CA" )&WORDS$( LEFT )&HEX$( "FE,
F7,CA" )&WORDS$( RIGH )&HEX$( "C9" )
590 CODE PRI=HEX$( "3A,9,2,F5,DD,E5,3E,1B,32,9,2,26,0,DD,6E,5,37,E,20,D7,
39,0,DD,E1,26,0,DD,6E,6,37,E,20,D7,39,0,F1,32,9,2,C9" )
600 CODE VARS=HEX$( "CD" )&WORDS$( BILL )&HEX$( "CD" )&WORDS$( PAP )&
HEX$( "CD" )&WORDS$( PRI )
610 CODE =HEX$( "CD" )&WORDS$( COLORS )&HEX$( "cd" )&WORDS$( SCL )
620 CODE =HEX$( "C9" )
630 CODE ATED=HEX$( "F3,DD,E5,3E,FE,D3,B1,DB,B3,F5,3E,FB,D3,B3,
FB,21" )&WORDS$( RTN )&HEX$( "CD" )&WORDS$( VARS )&HEX$( "0,21" )&WORDS$( RTN )&
HEX$( "CD" )&WORDS$( BOX1 )&HEX$( "3E,69,F7,5,78,0,21" )&WORDS$( RTN )
640 CODE =HEX$( "FE,A8,28,E5,FE,A4,28,D,FE,A0,28,E,FE,D,28,F,CD" )&
WORDS$( BOX2 )&HEX$( "18,D4,CD" )&WORDS$( BOX3 )&HEX$( "18,cf,CD" )&

```



```

WORDS(BOX4)&HEX$( "18,Ca,21" )&WORDS(RTN)&HEX$( "CD" )&WORDS(BOX2)&
HEX$( "F3,F1,D3,B3,3E,F9,D3,B1,DD,E1,FB,C9" )
650 CODE RST=HEX$( "11" )&WORDS(RST+6)&HEX$( "F7,1A,C9,5" )&"BASIC"
660 POKE 49144,REM(RST,256):POKE 49145,INT(RST/256)
670 LET A=PEEK(49140)+256*PEEK(49141)+8
680 POKE A,0:POKE A+1,31:POKE A+2,0:POKE A+3,0
690 CALL USR(KURSIV,0)
700 NUMERIC SD(1 TO 13)
710 LET DE,BC,X,Y,KC,FLG=0:LET XS=1:LET YS=9:LET FN$=""
720 CLEAR #102
730 REM CALL SCRIN("POSTER.SCR")
740 REM CALL ATRIN("POSTER.ATR")
750 REM DISPLAY #101:AT 4 FROM 1 TO SD(4)
760 PRINT #102,AT 2,16:"THAT'S IT!"
770 PRINT #102,AT 16,12:"ENTERPRISE UTILITY"
780 PRINT #102,AT 18,15:"VERSION 1.00"
790 PRINT #102,AT 23,6:"- PRESS ENTER BAR TO CONTINUE -"
800 INPUT #105:A$
810 REM CLOSE #101
820 REM CLOSE #27
830 LET DE,BC,X,Y,KC,FLG=0:LET XS=1:LET YS=9:LET FN$=""
840 CLEAR #102
850 SET #102:PALETTE 0,CYAN,0,YELLOW
860 SET 26,1
870 DISPLAY #102:AT 1 FROM 1 TO 24
880 PRINT AT 23,1:
890 PRINT #102,AT 5,8:"<<< POSTER MACHINE MENU >>>"
900 PRINT #102,AT 8,12:"1 - LOAD SCREEN"
910 PRINT #102,AT 10,12:"2 - LOAD ATTRIBUTES"
920 PRINT #102,AT 12,12:"3 - SAVE SCREEN"
930 PRINT #102,AT 14,12:"4 - SAVE ATTRIBUTES"
940 PRINT #102,AT 16,12:"5 - GO TO EDITOR"
950 PRINT #102,AT 18,12:"6 - HELP PAGE"
960 PRINT #102,AT 20,12:"7 - SAVE DEFINITIVE"
970 PRINT #102,AT 22,12:"8 - INITIAL"
980 GET #105:H$
990 SELECT CASE VAL(H$)
1000 CASE 1
1010 CALL FILENAME
1020 CALL SCRIN(FN$)
1030 GOTO 840
1040 CASE 2
1050 CALL FILENAME
1060 CALL ATRIN(FN$)
1070 GOTO 840
1080 CASE 3
1090 CALL FILENAME
1100 LET HEAD=1
1110 CALL SCROUT(FN$)
1120 GOTO 840
1130 CASE 4
1140 CALL FILENAME
1150 CALL ATROUT(FN$)
1160 GOTO 840
1170 CASE 5
1180 IF FLG=1 THEN CALL ATRED
1190 GOTO 840
1200 CASE 6
1210 CALL HELP
1220 GOTO 840
1230 CASE 7
1240 LET HEAD=0
1250 IF FLG=0 THEN 840
1260 CALL FILENAME
1270 LET XN$=FN$&".BIT"
1280 CALL SCROUT(XN$)
1290 LET YN$=FN$&".ATR"
1300 CALL ATROUT(YN$)
1310 GOTO 840
1320 CASE 8
1330 IF FLG=1 THEN
1340 CLOSE #27
1350 CLOSE #101
1360 END IF
1370 GOTO 830
1380 CASE ELSE
1390 !
1400 END SELECT
1410 GOTO 980
1420 DEF HELP
1430 CLEAR #102
1440 PRINT CHR$(246);

```

```

1450 PRINT #102,AT 1,5:"<<< POSTER MACHINE HELP PAGE >>>"
1460 PRINT #102,AT 3,2:">EDITOR"
1470 PRINT #102,AT 5,3:"INT JOYSTICK ---- MOVE"
1480 PRINT #102,AT 6,3:"SPACE & JOY ---- CURSOR SIZE"
1490 PRINT #102,AT 7,3:"PRESS A/S ----- SET PAPER"
1500 PRINT #102,AT 8,3:"PRESS D/F ----- SET INK"
1510 PRINT #102,AT 9,3:"1/Q ... 8/I ---- SET COLOURS"
1520 PRINT #102,AT 10,3:"PRESS 9 ----- SET BIAS"
1530 PRINT #102,AT 11,3:"<ERASE> KEY ---- DOWN PAPER ATTRS"
1540 PRINT #102,AT 12,3:"<DEL> KEY ----- DOWN INK ATTRS"
1550 PRINT #102,AT 13,3:"<INS> KEY ----- DOWN PAPER&INK ATTRS"
1560 PRINT #102,AT 14,3:"PRESS X/C ----- LEFT/RIGHT SCROLL"
1570 PRINT #102,AT 15,3:"<ENTER> ----- QUIT EDITOR"
1580 PRINT #102,AT 18,2:">FILES"
1590 PRINT #102,AT 20,3:"<F1> ---- PRINT 'LORI3'"
1600 PRINT #102,AT 21,3:"<F2> ---- PRINT 'TAPE:'"
1610 PRINT #102,AT 22,3:"<F3> ---- PRINT 'DISK:'"
1620 PRINT CHR$(246);
1630 PRINT #102,AT 24,11:"- PRESS KEY TO RETURN -"
1640 INPUT #105:A$
1650 END DEF
1660 DEF FILENAME
1670 SET 36,1
1680 SET 37,1
1690 SET 35,255
1700 PRINT AT 23,1:CHR$(161);
1710 PRINT AT 24,1:CHR$(161);
1720 INPUT #0,AT 23,1,PROMPT " >Filename? ":FN$
1730 PRINT AT 23,2:CHR$(161);">OK?";
1740 LET A$=INKEY$
1750 IF A$="" THEN 1740
1760 IF A$="N" OR A$="n" THEN 1670
1770 SET 26,0
1780 END DEF
1790 DEF ATRIN(AT$)
1800 IF FLG<>1 THEN GOTO 2890
1810 POKE BITLD+15,6
1820 POKE BITLD+9,REM(KC,256)
1830 POKE BITLD+10,INT(KC/256)
1840 POKE BITLD+12,REM(DE,256)
1850 POKE BITLD+13,INT(DE/256)
1860 OPEN #17:AT$ ACCESS INPUT
1870 CALL USR(BITLD,0)
1880 END DEF
1890 DEF SCROUT(ROU$)
1900 LET JK$=""
1910 IF FLG<>1 THEN GOTO 2890
1920 POKE BITLD+15,8
1930 POKE BITLD+9,REM(KC,256)
1940 POKE BITLD+10,INT(KC/256)
1950 POKE BITLD+12,REM(BC,256)
1960 POKE BITLD+13,INT(BC/256)
1970 OPEN #17:ROU$ ACCESS OUTPUT
1980 IF HEAD=0 THEN 2080
1990 FOR I=1 TO 4
2000 LET JK$=JK$&CHR$(SD(I))
2010 NEXT I
2020 FOR I=2 TO 9
2030 LET JK$=JK$&CHR$(PEEK(SEQ+I))
2040 NEXT I
2050 ASK 28 JK
2060 LET JK$=JK$&CHR$(JK)
2070 PRINT #17:JK$;
2080 CALL USR(BITLD,0)
2090 LET HEAD=0
2100 END DEF
2110 DEF ATROUT(AO$)
2120 IF FLG<>1 THEN GOTO 2890
2130 POKE BITLD+15,8
2140 POKE BITLD+9,REM(KC,256)
2150 POKE BITLD+10,INT(KC/256)
2160 POKE BITLD+12,REM(DE,256)
2170 POKE BITLD+13,INT(DE/256)
2180 OPEN #17:AO$ ACCESS OUTPUT
2190 CALL USR(BITLD,0)
2200 END DEF
2210 DEF ATRED
2220 DISPLAY #27:AT 27 FROM 1 TO 1
2230 DISPLAY #101:AT 1 FROM 1 TO SD(4)
2240 SET #102:PALETTE 0,0,0,0
2250 CALL USR(ATED,0)
2260 END DEF
2270 DEF SCRIN(SCR$)

```

```

2280 SELECT CASE FLG
2290 CASE 0
2300 LET FLG=1
2310 CASE ELSE
2320 GOTO 2930
2330 END SELECT
2340 OPEN #17:SCR$ ACCESS INPUT
2350 FOR A=1 TO 13
2360 GET #17:A$
2370 LET SD(A)=ORD(A$)
2380 NEXT
2390 IF SD(1)<>5 OR SD(2)<>0 THEN
2400 CLOSE #17
2410 LET FLG=0
2420 GOTO 2910
2430 END IF
2440 SET VIDEO MODE 15
2450 SET VIDEO COLOR 0
2460 SET VIDEO X SD(3)
2470 SET VIDEO Y SD(4)
2480 OPEN #101:"VIDEO:"
2490 FOR A=5 TO 12
2500 SET COLOR A-5,SD(A)
2510 POKE SEQ+A-3,SD(A)
2520 NEXT
2530 SET BIAS SD(13)
2540 LET BC=(65536+USR(BITMAP,0))-16384
2550 LET DE=(65536+USR(ATRMAP,0))-16384
2560 LET KC=SD(3)*SD(4)*9
2570 POKE BITLD+9,REM(KC,256):POKE BITLD+10,INT(KC/256):POKE BITLD+15,6
2580 POKE BITLD+12,REM(BC,256):POKE BITLD+13,INT(BC/256)
2590 POKE RTN+7,REM(DE,256):POKE RTN+8,INT(DE/256)
2600 POKE RTN+9,REM(BC,256):POKE RTN+10,INT(BC/256)
2610 POKE RTN+4,SD(3):POKE RTN+5,1:POKE RTN+6,2
2620 POKE RTN+2,XS:POKE RTN+3,YS:POKE RTN+11,33
2630 POKE RTN+0,0:POKE RTN+1,0:POKE RTN+12,SD(4):POKE RTN+13,SD(4)*9
2640 POKE RTN+14,REM(DE,256):POKE RTN+15,INT(DE/256)
2650 CALL USR(BITLD,0)
2660 POKE LEFT+1,REM(DE-1,256):POKE LEFT+2,INT((DE-1)/256):
      POKE LEFT+4,SD(4)*9:POKE LEFT+8,SD(3):POKE LEFT+24,SD(3)
2670 POKE RIGH+1,REM(BC,256):POKE RIGH+2,INT(BC/256):
      POKE RIGH+4,SD(4)*9:POKE RIGH+8,SD(3):POKE RIGH+24,SD(3)
2680 SET VIDEO X 8
2690 SET VIDEO Y 1
2700 SET VIDEO MODE 0
2710 SET VIDEO COLOR 0
2720 OPEN #27:"VIDEO:"
2730 SET #27:PALETTE 0,255,0,0
2740 SET #27:SCROLL OFF
2750 END DEF
2760 END WHEN
2770 !ERRORS!
2780 SOUND PITCH 60,DURATION 5
2790 SELECT CASE ERR
2800 CASE 1
2810 PRINT #102,AT 24,2:"*** Invalid LOR13 header."
2820 CASE 2
2830 PRINT #102,AT 24,2:"*** Select 'SCREEN LOAD' function."
2840 CASE ELSE
2850 PRINT #102,AT 24,2:"*** Select 'INITIAL' function."
2860 END SELECT
2870 WAIT 3
2880 EXIT DEF
2890 LET ERR=2
2900 GOTO 2770
2910 LET ERR=1
2920 GOTO 2770
2930 LET ERR=3
2940 GOTO 2770
2950 HANDLER LETSGO
2960 SOUND PITCH 60,DURATION 5
2970 PRINT AT 24,1:"*** USER ERROR! Poster restart."
2980 WAIT 3
2990 CALL USR(RST,0)
3000 END HANDLER

```

ISDOS betöltés

Az ISD_LOAD program az ISDOS automatikus betöltődésének kicsit kifinomultabb változatát valósítja meg. Az ISDOS szokványos és általam kidolgozott betöltési folyamata az ábrán látható. Az átalakítást mindössze az EXDOS.INI fájlban kell elvégezni, ahol az ISDOS helyére LOAD ISD_LOAD frandó. A program ekkor már működni fog, bár még nem látható a hatása, csak ha megírjuk a megfelelő batch fájlokat. Ezután a megadott billentyűk egyikének nyomvatartása a megfelelő batch fájlba irányítja a program futását.

A könnyebb érthetőség kedvéért nézzünk meg egy példát! A directory-ban a következő fájlok szerepelnek:

```
IS-DOS.SYS
EXDOS.INI
AUTOEXEC.BAT
ISD_LOAD
LTSH.BAT
```

Indításkor, ha nyomva tartottuk a bal Shift billentyűt, az LTSH.BAT, egyébként az AUTOEXEC.BAT fut le.

A lehetséges billentyű-fájl kombinációk a vázlatról vagy az assembly lista végéről leolvashatók.

A programot ASMON-ba kell beírni, majd ISD_LOAD néven 5-ös (NAP) fejléccel fordítani. Aki nem vállalkozik erre, az próbálja meg a Basic nyelvű betöltővel, amely a lapban meg szokott betöltőkhöz hasonlóan működik.

Az ISDOS.SYS, EXDOS.INI, ISD_LOAD valamint a megírt batch fájlokat célszerű /h (hidden) opcióval ellátni, így nem jelennek meg a directory listán.

Hámori György

```
100 PROGRAM "ISD_LOAD.BAS"
110 ALLOCATE 350
120 CODE ELEJE=HEX$("00,05,0E,01,00,00,00,
00,00,00,00,00,00,00,00,00,00,31,FF,3F,
FB,3E,01,D3,B5,DB,B5,CB,7F,CA,41,01,
3E,01,D3,B5,DB,B5,CB,4F,CA,51,01,3E,
00,D3,B5,DB,B5,CB,7F")
130 CODE =HEX$("CA,61,01,3E,08,D3,B5,DB,B5,
CB,6F,CA,71,01,3E,07,D3,B5,DB,B5,CB,
7F,CA,81,01,11,CA,01,C3,8E,01,11,B8,
01,3E,0A,F7,01,C2,3B,01,11,F3,01,C3,
8E,01,11,C1,01")
140 CODE =HEX$("3E,0A,F7,01,C2,3B,01,11,FF,
01,C3,8E,01,11,9E,01,3E,0A,F7,01,C2,
3B,01,11,00,01,C3,8E,01,11,A7,01,3E,
0A,F7,01,C2,3B,01,11,DC,01,C3,8E,01,
11,B0,01,3E,0A")
150 CODE =HEX$("F7,01,C2,3B,01,11,EB,01,05,
3E,0A,F7,03,01,F7,1A,11,0B,02,F7,1A,
C3,9B,01,0B,4C,54,53,4B,2E,42,41,54,
0B,52,54,53,4B,2E,42,41,54,07,41,4C,
54,2E,42,41,54")
160 CODE =HEX$("08,43,54,52,4C,2E,42,41,54,
0B,4C,4F,43,4B,2E,42,41,54,05,69,73,
64,6F,73,0B,69,73,64,6F,73,20,2F,4C,
54,53,4B,0B,69,73,64,6F,73,20,2F,52,
54,53,4B,0A,69")
170 CODE =HEX$("73,64,6F,73,20,2F,41,4C,54,
0B,69,73,64,6F,73,20,2F,43,54,52,4C,
0B,69,73,64,6F,73,20,2F,4C,4F,43,4B,
02,57,50,00,0A,00,00,00,00,00,00,00,
00,00,00,00,00")
180 CODE =HEX$("00,00")
190 CODE VEGE=HEX$("0")
200 OPEN #10:"DISK:ISD_LOAD" ACCESS OUTPUT
210 LET S=0
220 FOR I=ELEJE TO VEGE-1
230 PRINT #10:CHR$(PEEK(I));
240 LET S=S+PEEK(I)
250 NEXT I
260 CLOSE #10
270 IF S<>24865 THEN
280 PRINT "**** ISD_LOAD: Hiba az adatokban!!!"
290 ELSE
300 PRINT "ISD_LOAD rendben!"
310 END IF
```

```
; ISDOS szelektív betölto
; Hamori Gyorgy 1992.04.05.
BILL MACRO OutByte,InPos
ld a,OutByte
out(181),a
in a,(181)
bit InPos,a
ENDM
```

```
ORG 100h
ld sp,3fffh
ei
Bill 1,7
jp z,CTRL
Bill 1,1
jp z,LOCK
Bill 0,7
jp z,LTSH
Bill 8,5
jp z,RTSH
Bill 7,7
jp z,ALT
```

```
NOT_F ld de,M_NOT
jp HIV
```

```
CTRL ld de,F_CTRL
ld a,10
exos 1
jp nz,NOT_F
ld de,M_CTRL
jp HIV
```

```
LOCK ld de,F_LOCK
ld a,10
exos 1
jp nz,NOT_F
ld de,M_LOCK
jp HIV
```

```
LTSH ld de,F_LTSH
ld a,10
exos 1
jp nz,NOT_F
ld de,M_LTSH
jp HIV
```

```
RTSH ld de,F_RTSH
ld a,10
exos 1
jp nz,NOT_F
ld de,M_RTSH
jp HIV
```

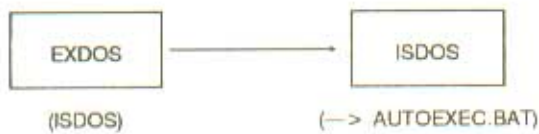
```
ALT ld de,F_ALT
ld a,10
exos 1
jp nz,NOT_F
ld de,M_ALT
```

```
HIV push de
ld a,10
exos 3
pop de
exos 26
ld de,M_WP
exos 26
```

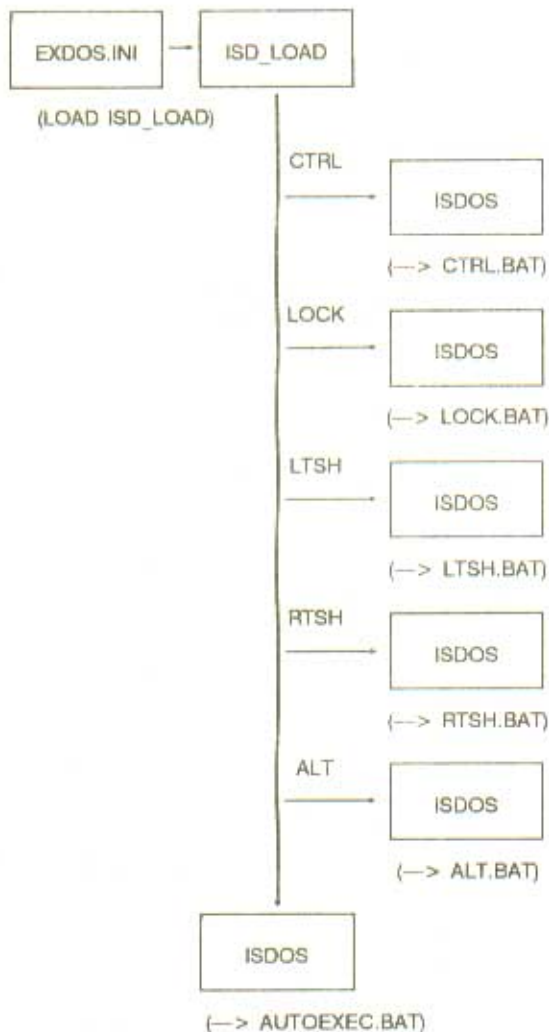
```
CC jp CC
```

```
F_LTSH DB 8,"LTSH.BAT"
F_RTSH DB 8,"RTSH.BAT"
F_ALT DB 7,"ALT.BAT"
F_CTRL DB 8,"CTRL.BAT"
F_LOCK DB 8,"LOCK.BAT"
M_NOT DB 5,"isdos"
M_LTSH DB 11,"isdos /LTSH"
M_RTSH DB 11,"isdos /RTSH"
M_ALT DB 10,"isdos /ALT"
M_CTRL DB 11,"isdos /CTRL"
M_LOCK DB 11,"isdos /LOCK"
M_WP DB 2,"WP"
END
```

Az ISDOS szokványos betöltésének vázlatja:



A módosított betöltés vázlatja:



Megj.: Ha a megadott BATCH fájl nem létezik, az ISDOS paraméter nélkül kerül betöltésre. Ha ez sem sikerül, a WP töltődik be.

Sprite kezelés ENTERPRISE-on

A sprite-ok, másnéven szellemek vagy manók, a számítógépes játékok legfőbb alkotóelemei, ezért azokban a gépekben, amelyeket főleg játékokra terveztek egy külön hardver gondoskodik a mozgásukról. Sajnos az ENTERPRISE nem büszkélkedhet ilyen képességekkel, tehát kénytelenek vagyunk programból megvalósítani ugyanezt. Megpróbálok néhány BASIC és ASSEMBLY rutin segítségével bemutatni a sprite-kezelés módszereit.

Az első és egyben legprimitívebb eljárás (amelyet általában a kezdő BASIC-programozók alkalmaznak előszeretettel), hogy kirajzoljuk a háttérrel majd erre kirakjuk a manókat az aktuális pozíciójukba. Ezután elvégezzük a többi szükséges műveletet (billentyűzetfigyelés, ellenőrzések, számítások stb.), majd letöröljük a képernyőt és kezdjük előlről az egészet. Ez a módszer nagyon lassú, ezért az egész kép villog, még hozzá nem is kis mértékben. De mivel az ENTERPRISE-nak olyan remek videóchip-je van, mint a NICK, ezért még ezt a legősibb eljárást is megoldhatjuk villogásmentesen, mégpedig egy háttérképernyő segítségével. A lényeg, hogy két, méreteiben azonos videólapot nyitunk meg, amelyek közül mindig csak az egyiket kapcsoljuk be, és a azon dolgozunk, amelyik nem látszik (ezt hívjuk háttérképernyőnek). Miután mindent kiraktunk rá, egy DISPLAY utasítással átkapcsolunk erre a videólapra. Ha mindent jól csinálunk a képernyőn semmi nem fog villogni, azonban a programunk nagyon lassú és ezért élvezhetetlen lesz, ezért ezt a módszert csak olyanoknak ajánlom, akik most ismerkednek a programozással.

A második lehetőség, hogy a sprite-ok szélein egy vagy több pixel szélességű üres sávot hagyunk. Ha ezt úgy tesszük ki a képernyőre, hogy az előző fázis helyétől csak annyi képponttal tér el az új pozíció ahány egység szélességű üres sáv van a szélein, akkor ez teljesen letörli az előző fázis egyébként kilógó részét. Mint látható ennél a módszernél nincs külön kirakás és törlés, hanem mindkettő egyszerre hajtodik végre. Ezért ez az eljárás kellően gyors ahhoz, hogy akár BASIC-ben is alkalmazhassuk. Legnagyobb hátránya, hogy nem lehet a szellem mögött háttér mert mozgása közben azt is letörölné, a másik hibája, hogy csak kis egységenként lehet mozgatni (maximum annyi ponttal amilyen széles az üres sáv a szélén). Ezen hiányosságai ellenére BASIC-ben a legjobb megoldás a sprite-kezelésre, persze csak akkor ha nem alkalmazunk háttérrel. Sajnos az ENTERPRISE BASIC-ben ezt a sprite kezelési módszert nem vagy csak nagyon körülményesen lehet megvalósítani, ugyanis ha grafikus lapra írunk a PRINT utasítással akkor az nem írja felül azt, ami kint van a képernyőn, hanem hozzámásolódik. Ezért a manó szélein hagyott keret nem fogja letörölni az előző fázist.

A következő megoldás, amikor a manót úgy rakjuk ki, hogy össze XOR-oljuk a képernyőn lévő háttérrel. Mivel ennek a logikai műveletnek az a tulajdonsága, hogy ha egy értéket kétszer össze XOR-olunk egy másikkal, akkor visszakapjuk az eredetit, ezért a sprite letörlése nagyon egyszerű, csak újra ki kell raknunk. A leírtakból már látszik, hogy ismét két fázisból áll a művelet, tehát lassabb lesz, mint az előző, de itt nyugodtan mozoghat a manó egy háttér előtt is, nem fogja azt letörölni. Természetesen a XOR művelet miatt színes képernyőn a végeredmény nem mindig a legszebb. Magasszintű programnyelvekben ez a legelterjedtebben alkalmazott módszer. Egy BASIC nyelvű

példaprogram az 1. listán látható, ami alapján remélem mindenki számára világos lesz az eljárás.

A legjobb és gépi kódú programokban általánosan használt eljárás az ún. maszkolt sprite kezelés. Itt az egyes fázisok két részből állnak, az egyik maga a megrajzolt alak, a másik pedig ennek a maszkja. A maszkot úgy készítjük el, hogy az alakzat körvonalán kívül eső pontok bitjeit kigyűjtjük a többit pedig kinullázzuk. A módszer lényege a következő: a képernyő azon részéről - ahová a sprite-ot ki akarjuk tenni - elmentjük az adatokat egy átmeneti ún. puffer-tárba, majd a képernyő megfelelő bájtyát le AND-eljük a maszk megfelelő bájtyával és ehhez hozzá OR-oljuk a sprite ugyanazon bájtyát, az eredményt pedig visszateszük a képernyőre. A következő kirakás előtt pedig a pufferből visszamásoljuk a képernyő eredeti tartalmát. A jobb érthetőség érdekében a 2. listán látható egy példa a módszer megvalósítására. A TOROL címkejű rutin végzi az előző fázis törlését, a KIRAK címkejű pedig a kirakást. Ez utóbbinak a HL-ben kell átadni azt a képernyőcímet, ahová a manót akarjuk tenni (bal felső bájtyának a címe), természetesen hívása előtt be kell lapozni a megfelelő videószegetmenst. Ezekkel a rutinokkal csak egy sprite mozgását tudjuk elvégezni, de bárki könnyen átalakíthatja őket. Több sprite használata esetén figyelni kell arra, hogy a törlésnél először az utoljára kirakottat kell törölni, majd az előzőt és utoljára azt, amelyiket először raktunk ki és csak az összes szellem letörlése után kezdhethetjük kirakni őket újra. Ha nem így csináljuk, akkor nem mehetnek egymásra a manók, mert összezavarodik a képernyő. Ha viszont így csináljuk, akkor sokáig nem lesznek kint a sprite-ok a képen, ezért villogni fognak. Ezt kiküszöbölhetjük úgy, hogy a sprite kezelést a videó megszakításban végezzük. Azonban ha nagyon sok manót akarunk használni akkor csak a háttérképernyő használata vezet eredményre, amelyről már volt szó.

Remélem mindenki megtalálta a neki legjobban tetsző sprite kezelést, és sok jó, villogásmentes játékkal fog gazdagodni az ENTERPRISE-tábor.

Devil

Ilyen még a neppereknél sincs...

SPRED release 1.5

Felhasználóbarát Entersprite kompatibilis sprite editor

- Tömértelen funkció
- Pull-down menürendszer
- Esztétikus kivitel
- Exdos használat
- Beépített help
- Magyar nyelvű .WP leírás

Mindez gyorsan, gépi kódban!

Ára csak 299 Ft!

Befizetésedet rózsaszínű postautalványon várjuk. Ha nem küldesz 5.25"-os lemezt/kazettát, akkor még 40 Ft-ot adj az árhoz. A postaköltség a program árában benne van.

Cím: ARSS, 1399 Budapest, Pf. 701/334.

SPRED r1.5 ... és leesik az álad.

Vége a Sinclair uralomnak!

Megalakult az Országos ENTERPRISE Klub.

Mindenki jelentkezzen, aki lépést akar tartani gépe fejlődésével!

Kérje részletes tájékoztatónkat válaszborítékkal!

**Tagtoborzó: Silye Gabriella
5358 Tiszaörvény, Rákóczi út 4.**

```
100 PROGRAM "SPRITE.BAS"
200 GRAPHICS HIRES 16
120 SET KEY CLICK OFF
130 SET LINE MODE 3
140 LET SPRX=10:LET SPRY=5:LET E$=CHR$(154)
150 PRINT #101,AT SPRY,SPRX:E$
160 LET LSPRX=SPRX:LET LSPRY=SPRY
170 LET JS=JOY(0)
180 LET SPRX=SPRX-((JS BAND 1)=1 AND
    SPRX<20)+((JS BAND 2)=2 AND SPRX>1)
190 LET SPRY=SPRY-((JS BAND 4)=4 AND
    SPRY<10)+((JS BAND 8)=8 AND SPRY>1)
200 IF LSPRX=SPRX AND LSPRY=SPRY THEN 170
210 PRINT #101,AT SPRY,SPRX:E$
220 PRINT #101,AT LSPRY,LSPRX:E$
230 GOTO 160
```

1. lista

```
.RADIX 16
SPRX EQU 2 ;SPRITE VIZSZ.MERETE BYTEBAN
SPRY EQU 10 ;SPRITE FUGG.MERETE PONTBAN
LINESIZE EQU 50 ;VIDEOLAP VIZSZ.MERETE
SPRITEDAT EQU 4000 ;SPRITE MEMORIACIME
MASKDAT EQU 4020 ;MASZK MEMORIACIME

TOROL LD HL,(LASTSPR) ;AZ ELMENTETT
LD DE,PUFFER ;KEPERNYOADATOK
LD A,SPRY ;VISSZAMASOLASA

SPROFF PUSH HL
LD BC,SPRX
EX DE,HL
LDIR
POP BC
EX DE,HL
LD HL,LINESIZE
ADD HL,BC
DEC A
JP NZ,SPROFF
RET

KIRAK LD (LASTSPR),HL
DE,HL
LD HL,SPRITEDAT ;HL-BE SPR CIM
EXX
LD HL,MASKDAT ;HL'-BE MASZK CIM
LD BC,PUFFER ;BC'-BE A PUFFER CIM
EXX
LD B,SPRY
SPRON2 PUSH BC
PUSH DE
LD B,SPRX
SPRON LD A,(DE) ;A-BA A KEP MEGF. BAJT
EXX
LD (BC),A ;ELTAROLASA A PUFFERBE
AND (HL) ;LE AND-ELESE A MASZKKAL
INC BC
INC HL
EXX
OR (HL) ;A SPRITE HOZZA OR-OLASA
LD (DE),A ;VISSZAIRAS A VIDEOM.
INC HL
INC DE
DJNZ SPRON ;CIKLUS ISMETLESE
POP BC ;X MERETETOL FUGGOEN
EX DE,HL
LD HL,LINESIZE ;A KOV. SOR
ADD HL,BC ;CIMENEK KISZAMITASA
EX DE,HL
POP BC
DJNZ SPRON2 ;CIKL ISMETL.A SPR
RET ;Y MERETETOL FUGGOEN

LASTSPR DEFW 0 ;UTOLJARA KIRAKOTT FAZ.
PUFFER DEFS SPRX*SPRY ;PUFFER
```

2. lista

Memóriabővítés 320 KB-ra

ENTERPRISE-unk 128 kilobájtos RAM (olvasható és írható) memóriája nem mondható kevésnek a hasonló kategóriájú házi számítógépekhez viszonyítva. A Z80-as "szabvány" 64 kilobájtot használ, hiszen gépünk "agya" ennyi memóriát tud közvetlenül elérni. Az ENTERPRISE megalkotóit illeti a dicséret azért, hogy összesen négy megabájt lett a gépünk által címezhető tartomány. Elvileg tehát nincs akadálya a bővítésnek.

Bizony előadódhatnak olyan helyzetek hogy az alap-memória kevésnek bizonyul. Ha több EXOS-rendszerbővítőt töltünk be, egyenként lefoglalják a maguk tizenhat kilobájtját, és ezt egészen a gép kikapcsolásáig, vagy hirtelintésig teszik. Nem is beszélve az olyan ötletekről, amikor RAMDISK-nek, azaz látszólagos lemezegységnek akarjuk használni a memória egy részét. Ennek nagy előnye hogy egy puszok gyors "floppy-meghajtóhoz" jutunk programok és adatok átmeneti tárolására, hátránya hogy "zabálja" a memóriát.

A következőkben az ENTERPRISE belső memóriájának 320 kilobájtra bővítését ismertetjük. A működés végigkövetéséhez nem árt ha kéznél van a gép kapcsolási rajza. Az átalakítás lényege hogy az eredetileg 2×64 kilobájtos RAM-csoport közül az egyiket 256K-ra cseréljük. Így jön ki a $64 + 256 = 320K$! Az alappanelon lévő 64 kilobájtot nem bántjuk, hanem csak a beépített bővítőpanelba forrasztott nyolc darab 4164 típusú $64K \times 1$ bites chipet cseréljük ki szintén nyolc darab 41256 típusúra. Ezek $256K \times 1$ bitesek, tehát négyszer akkora kapacitásúak. Kapcsolásuk az 1. ábrán látható.

A dinamikus RAM-okra jellemző multiplexelt címzés (egy lábón időben eltolva, azaz egymás után kétféle címet kap az IC) következtében a 4164-es $2^8 = 16$ címbit igényel. Az egyes láb nines bekötve. A 41256-nál $2^9 = 18$ címbit aktív. A többi kivezetés megegyezik. Az első probléma a megnövekedett címtartománynak megfelelő címkiválasztás biztosítása.

Ezt a feladatot oldja meg az 1. ábra bal oldalán lévő kapcsolás. A 74LS151-es multiplexerre kell vezetni az A16, A17, A18 címeket. Kimenete a szintén címszelektálást végző U110 2. lábára megy a bővítőpanelon.

Átalakítások: U112 (LS00) 2. lábáról lekötni A16-ot, mert oda föld kerül. U110 1. lábáról lekötni A17-et, mert oda +5V kerül; 2. lábáról pedig lekötni A18-at, mert oda az említett multiplexer 5. lába kerül. Az SW-jelű kapcsoló bekapcsolás állapotában a hagyományos 128 kilobájt áll rendelkezésre, kikapcsolva pedig 320K, mégpedig folyamatosan elfoglalva az 1-Ch-Flh szegmenseket.

További átalakítások: A beépített RAM-ok 1-es lábait összekötjük egymással, valamint az U109 (74F157) címmultiplexer 12. lábával. Ez utóbbit viszont lekötjük a RAM-ok 9. lábáról (A7).

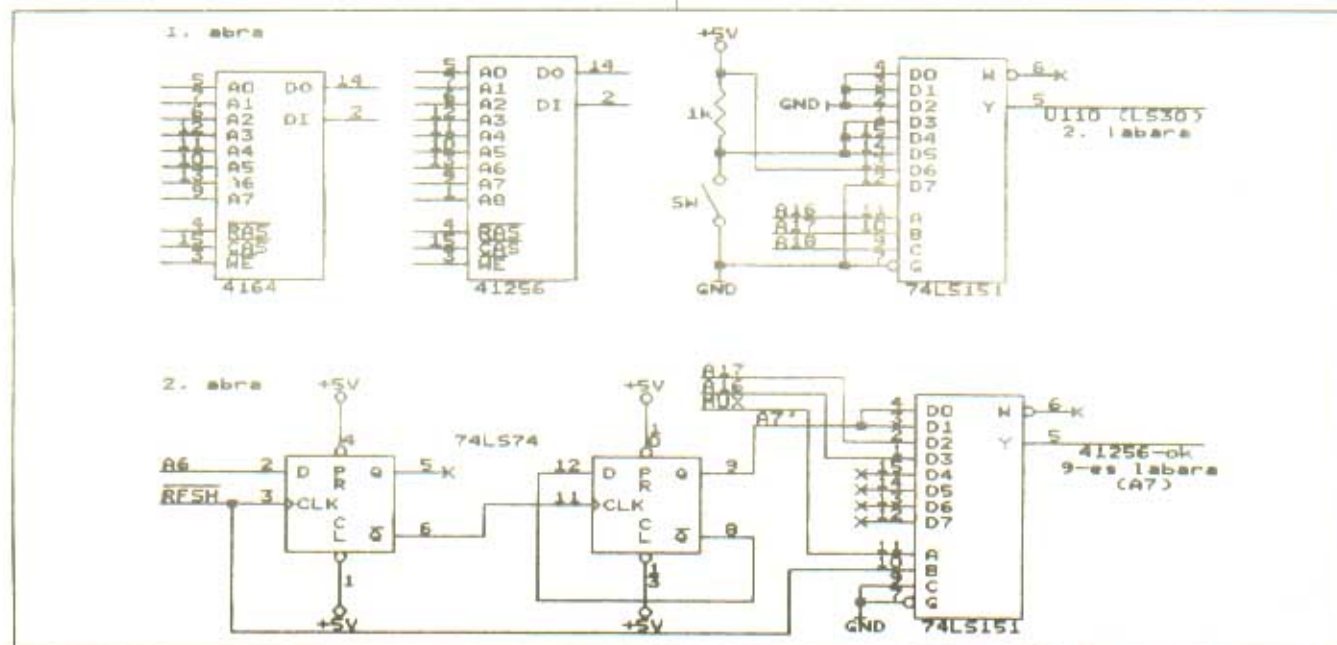
Az IC cserével felmerülő másik gond a RAM-frissítés megoldása. Ismert a tény, hogy a sor-oszlop szervezésű dinamikus RAM-nál legalább 2 millisecundumként minden sort meg kell címezni, azaz frissíteni kell, mert különben elveszhet a tartalom. A Z80 processzor a $64K$ -s DRAM-okig észrevétlenül megoldja ezt a feladatot hétféle frissítőregiszterével, a 41256-nak viszont eggyel több frissítő-bite van szüksége! Ezért egy trükkhöz kell folyamodnunk. A frissítésenként eggyel növekvő tartalmú frissítőregisztert kell megtölteni egy bittel. A 2. ábra kapcsolása azt eredményezi, hogy a legnagyobb helyértéket képviselő A6 frissítőbit alacsonyra váltásakor (persze csak aktív-RFISH jelnél) áthillen a jobboldali flip-flop A7-jelű kimenete. Frissítési ciklus alatt ez a jel kerül az LS151 multiplexer jövőlából a RAM-ok 9-es (A7) lábára. Egyébként az A16/A17-et kapjuk multiplexelve. Hogy mikor melyiket, az a MUX jelől függ, amely időben a -RAS és a -CAS közötti van. A bővítőpanelon lévő U108 (vagy U109) multiplexer 1. lábáról vehető le a MUX jel.

A beépítendő három TTL IC a bővítőpanelon lévőkre építhető rá. A RAM-okat viszont a régiek eltávolítása (kicsipdése) után foglalható ajánlatos ültetni. Még éppen el fognak férni a gép burkolata alatt. Az átalakítást csak kellően felkészült és háttér IP-rajongóknak ajánlom, hiszen mélyen bele kell nyúlnunk kedvelt gépünk "lelki világába", ehhez pedig jól kell ismerni a "páciens"!

Megjegyzem hogy találkoztam már olyan memóriabővítéssel, amelyik nem foglalkozott az utóbb ismertetett frissítés megoldásával, és csodák-csodájára működött is! Am felelősségem teljes tudatában kijelentem, hogy ez csak a véletlen (és a néha kifurkaszthatatlan elektronika) műve volt. Ugyanaz a kapcsolás másik gépen - kellő idejű hemelegedés után - rendszeres "el-szállási" produkált!

Finnyi lett volna a bővítés elvének leírása, ami nem annyira gyakorlati útmutató, hanem inkább gondolatébresztő akart lenni a hardver terén jártas "márkatársak" számára, hogy még jobban megismerjék gépüket.

Bozai Gábor



Ha minden fólia szakad...

(folytatás)

... Miután a billentyűzet-illesztő hardver oldalát sikerült ismertetni (elégké kimerítően), rátérhetünk a dolog felkének tárgyalására, a vezérlőprogramra.

A program két fő részből áll: egy alacsony szintű (a hardvert közvetlenül kezelő, ha valaki nem erre gondolt volna) és egy magasszintű (az EXOS felé kapcsolatot tartó) modult foglal magába. Ezek a listában is elkülönülnek egymástól. Tehát, ha valaki saját EXOS eszközt kíván definiálni, a kezelő rutinokat külön is felhasználhatja (vagy írhat helyette jobbat).

A forrás a HISOFT GEN ASSEMBLER szintaktikájának felel meg, de kis módosítással az ASMÓN (SIMÓN, turbó ASMÓN etc.) is le tudja fordítani. Az eltéréseket a kellő helyen megemlítem. A sor elején található szavak címkék, ezek elé nem szabad szóközt tenni. A szimbólumok után nem tettem ki a kettőspontot, mivel a GEN ezt nem szereti, az ASMÓN-nak meg mindegy. A megjegyzések (pontosvessző után) csak a megértést hivatottak segíteni, tehát elhagyhatók. Terjedelmi gondok miatt a teljes lista helyett csak a fontosabb részleteket választottam ki.

Ha valakiben maradt még vállalkozókedv mindezek után, akkor vágjunk bele!

Gyányi Sándor

```

;Az ASMÓN-nal dolgozók ezt a makrót hagyják ki!
EXOS   MACRO @
      RST 30H
      DEFB @
      ENDM

BREAKCODE EQU 128 ;A Legmagasabb bit maszkja (felengeds)

DATAPORT EQU 60H ;Adatkád portcáme
STATPORT EQU 61H ;"Kád rkezett" regiszter portcáme
RESPORT EQU 62H ;Alap lllapotba helyezs portcáme
;A fenti portcámek leár sa az elz rszben tal lthatá.

SHIFT EQU 1 ;A "SHIFT" jelz maszkja
CTRL EQU 2 ;A "CTRL" jelz maszkja
ALT EQU 4 ;Az "ALT" jelz maszkja

      ORG 0C00AH ;A bvát kezdécáme

;Belső pont
;Az eredeti "KEYBOARD:" perifria tdefiní l sa.

      LD A,C ;A "C" regiszterben az akciákád
      CP B ;Ha C=B, akkor inicializ l s
      LD A,0 ;Egybknt visszatrs
      RET NZ
      PUSH BC
      IN A,(0B3H) ;A t bl zat szegmenssz ma
      LD (00SEG),A ;t rolva a perifrialeárában

      CALL RESET ;Billentyűzet alap lllapotba

      LD DE,DDTYPE ;Eszkz rendszerbe fzsze
      LD BC,0 ;DE=perifrialeárÁ, BC=RAM mrete
      EXOS 21 ;Eszkz bel ncol s
      POP BC
      RET ;Visszatrs

;Eszkz leárÁ t bl zat
;A mezk nagyrat t ld. EXOS kziknyv
DDNEXT DEFW 0
DDNEXS DEFB 0
DDTYPE DEFB 0
DDIRQ DEFB 32
DDFLAGS DEFB 0
DDTAB DEFW TABLE-8000H ;Belső pontok az 1. lapon
D0SEG DEFB 0
DDUNIT DEFB 0
DDNAME DEFB 8
      DEFM "KEYBOARD"

;Belső pontok
;Magyar zatuk az EXOS knyvbén
TABLE DEFW IT
      DEFW OPEN
      DEFW CREATE
      DEFW CLOSE
      DEFW CLOSE
      DEFW READC
      DEFW READB
      DEFW WRITEC
      DEFW WRITEB

```

```

DEFW READS
DEFW WRITES
DEFW SFUNC
DEFW INIT
DEFW MOVEB

OPEN LD A,(ENABLE) ;Csatorna megnyit sa
      OR A ;Eszkz haszn lva?
      LD A,DE9H
      RET NZ ;igen, hibazenet
      CALL INIT
      XOR A
      LD (READY),A ;Pufferek trise
      LD (BYTE),A
      DEC A
      LD (ENABLE),A ;Haszn lat jelzse

      LD DE,0 ;Csatornapuffer ignyls
      EXOS 27
      RET

CREATE EQU OPEN ;Csatorna létrehoz sa

CLOSE JP INIT ;Csatorna lez r sa

READC LD A,(BYTE) ;Karakter olvas sa
      OR A
      JR NZ,READ1 ;Van karakter elklds alatt
      LD A,(READY) ;Van karakter a pufferben?
      OR A
      JR Z,READY2 ;Ha nincs, v rakoz s
      LD (BYTE),A ;Ha van, akkor elklds be llát sa
      LD HL,(CHAR1)
      LD (CHAR),HL
      XOR A
      LD (READY),A
      LD A,(BYTE)
      DEC A
      LD (BYTE),A
      LD HL,(CHAR)
      LD B,(HL) ;A karakter "B"-ben
      INC HL
      LD (CHAR),HL
      XOR A
      RET

READB LD A,B
      OR C
      RET Z
      PUSH BC
      PUSH DE
      CALL READC
      POP DE
      CALL PAGE
      LD (HL),B
      INC DE
      POP BC
      DEC BC
      LD A,B
      OR C
      JR NZ,READB1
      RET

READB1

PAGE LD A,D
      RLCA
      RLCA
      OR 0FCH
      LD L,A
      LD H,0BFH
      LD A,(HL)
      OUT (0B1H),A
      LD H,0
      LD L,E
      SET 6,A
      RES 7,H
      RET

WRITEC LD A,0E7H ;Karakter ár sa
      RET ;hibazenet

WRITEB EQU WRITEC

WRITES EQU WRITEC ;String ár sa

READS LD C,0 ;St tusz olvas sa
      LD A,(BYTE)
      OR A

```



```

ONE      JR  NZ,ONE
        LD  A,(READY)
        OR  A
        RET Z          ;Nincs olvasható karakter
        DEC C
        XOR A
        RET

```

```

SFUNC   LD  A,B          ;Spec. funkció hívása
        SUB B
        JR  C,ISPEC     ;Hibazenet
        JR  Z,FKEY
        CP  Z
        JR  C,JOY
        LD  A,OEAH
        RET

```

```

FKEY    LD  A,C          ;Funkcióbillentyű programozása
        AND 15
        LD  L,A
        LD  H,0
        ADD HL,HL
        ADD HL,HL
        LD  C,L
        LD  B,H
        ADD HL,HL
        ADD HL,BC
        LD  BC,FKEYS
        ADD HL,BC
        LD  A,(DE)
        CP  24
        LD  C,A
        LD  A,OE2H
        RET NC
        EX  DE,HL
        INC C
        LD  B,0
        LDIR
        XOR A
        RET

```

```

JOY     DEC C          ;JOYSTICK beolvasása
        JP  M,INT
        JR  Z,EXT1
        LD  C,5

```

```

EXT1    LD  B,C
        LD  C,8

```

```

EXT2    LD  A,(0BFF3H)
        AND 0F0H
        OR  B
        INC B
        OUT (0B5H),A
        IN  A,(0B6H)
        RRA
        CCF
        RL  C
        JR  NC,EXT2
        XOR A
        RET

```

```

INT     LD  C,1        ;Belső JOYSTICK
        XOR A
        RET

```

```

IT      LD  A,(ENABLE)
        OR  A
        RET Z          ;Nincs megnyitott csatorna
        LD  A,(READY)
        OR  A
        RET NZ        ;Teli a puffer
        CALL INPUT    ;Karakter beolvasása
        RET Z          ;Nincs karakter
        LD  (KEY),A
        CALL CLICK
        LD  HL,KEY
        LD  (CHAR1),HL
        LD  BC,9        ;KEYIRQ
        EXOS 16
        INC 0
        DEC 0
        JR  NZ,IT1
        LD  A,21H        ;SOFTWARE IT
        LD  (0BFF2H),A
        LD  A,(FUNC)    ;Funkcióbillentyű?
        BIT 7,A
        JR  NZ,IT2     ;Nem
        LD  L,A
        LD  E,A
        LD  H,0

```

```

        ADD HL,HL
        ADD HL,HL
        ADD HL,HL
        LD  C,L
        LD  B,H
        ADD HL,HL
        ADD HL,BC
        LD  BC,FKEYS
        ADD HL,BC      ;Funkcióbillentyű programozott szvege
        LD  A,(HL)
        OR  A
        JP  Z,SOFTIR    ;Ha nullstring, SOFTWARE IT
        LD  (READY),A   ;Klínben string elküldése
        INC HL
        LD  (CHAR1),HL
        RET

```

```

SOFTIR  LD  A,E
        ADD A,10H
        LD  (0BFF2H),A  ;Funkcióbillentyű IT jelzése
        XOR A
        RET

```

```

IT2     LD  A,(KEY)
        CP  3          ;STOP billentyű?
        JR  NZ,IT4
        LD  BC,8        ;Igen, STOPIRQ vizsgálat
        EXOS 16
        LD  A,0
        OR  A
        JR  NZ,IT4
        LD  A,20H        ;STOP IT jelzése
        LD  (0BFF2H),A
        RET

```

```

IT4     LD  A,1        ;Egy karakter jelzése
        LD  (READY),A  ;(CHAR1) m r beállítás
        RET

```

;Itt egy kis ugrás következik, a hiányzó rutinokat
;mindenképp észlelni kell.
;Az EXOS könyvben megtalálhatók az egyes funkciókhoz
;tartozó feladatok, paraméterek.

;A hardvert kezelő rutinok

;Billentyűkád beolvasása

```

SCAN    IN  A,(STATPORT);Adat olvasása a statuszportról
        AND 128        ;"Adat érkezett" jelzésziket
        RET Z          ;Visszatérés, ha nincs adat
        IN  A,(DATAPORT);Adat beolvasása
        RET

```

;Billentyűzet alaphelyzetbe állítása

```

RESET   DI          ;Megszakítás tiltása az időzítés miatt
        LD  E,10      ;Maximum 10 próbálkozás engedélyez
        IN  A,(RESPORT);Az árajelvezeték 0-ba hívása
        LD  BC,7000   ;Kb. 50ms várakozás
        RES2  DEC BC
        LD  A,B
        OR  C
        JR  NZ,RES2

```

```

        IN  A,(RESPORT);Az árajelvezeték felengedése
        LD  BC,200    ;1.35ms várakozás

```

```

RES3    DEC BC
        LD  A,B
        OR  C
        JR  NZ,RES3
        CALL SCAN     ;Statisztika beolvasása
        JR  NZ,RES4   ;Ha nem érkezett statisztika
        RES5  DEC E    ;Éjra próbálkozás
        JR  NZ,RES1

```

```

LOOP    LD  A,R        ;Billentyűzethiba, végtelen ciklus
        OUT (BTH),A   ;Miközben a keret villog
        JR  LOOP

```

```

RES4    CP  0AAH      ;OK 84gombos billentyűzetnél
        JR  Z,OKE     ;Helyes működés esetén visszatérés
        CP  55H       ;OK 103gombos billentyűzetnél
        JR  NZ,RES5   ;Ha nem ez érkezett, éjra próbálkozás
        OKE  EI        ;Helyes működés esetén visszatérés
        RET

```

PRINT-PACK: Nyomás!

PRINT-PACK néven új nyomtatásillesztő programcsomag került szerkesztőségünkhöz. A HSOFT névvel fémjelzett lemezen 19 fájl található. Köztük van négy EXOS rendszerbővítő. Ezek a HPRINT, a DDUMP, a KP és a VLOAD. Sorra vesszük mindet, hogy képet kaphassunk a sokat ígérő és sokakat érintő problémák e lehetséges megoldásáról.

A lemez egy rejtett EXDOS.INI fájjal indít, betölti a HPRINT-et, a DDUMP-ot majd a PRINT.SCR nevű demóképet, amit be is mutatunk olvasóinknak. Ezután automatikusan a beépített szövegszerkesztőbe kerülünk. Hogy miért, az

billentyű sorában keretező karaktereket is kapunk. A HPRINT help szövege elárulja, hogy E:PERSON kompatibilis ékezetes nyomtatóbővítés van birtokunkban, mely lehetővé teszi az ékezetes nyomtatást. A HPRINT.INI fájl tartalmazza a szöveghe illeszthető vezérlőkódoknak megfelelő szekvenciákat, amelyek vezérlik a nyomtatónkat. A HPRINT.INI-t szövegszerkesztővel tetszés szerint módosíthatjuk, ha nyomtatónk más kódokkal működne.

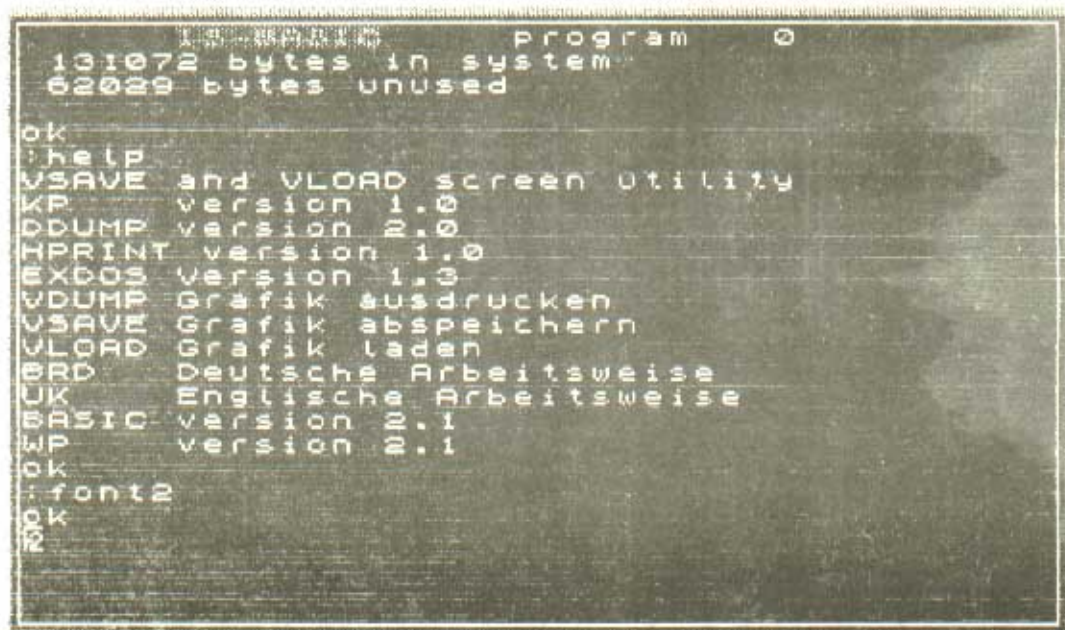
Nézzük a további EXOS parancsokat, melyek a HPRINT betöltése után élnek! A HFONT a képernyő és a billentyűzet



DDUMP version 2.0 HSOFT 1991.

sajnos rejtély marad előttünk. De sebjaj, mert vígan ugrándozhatunk a BASIC, a WP és a többi rendszerbővítés között. Ikkor már megjeleníthetők a képernyőn a magyar ékezetes karakterek. A [Q] billentyű sorában a nagybetűsek, az [A] sorában pedig a kisbetűsek esalogathatók elő az [Alt] segítségével. Az igazsághoz hozzátartozik, hogy logikusabb billentyűzetkiosztást is el tudnánk képzelni, de ne elégedetlenkedjünk hiszen a [Z]

ékezetesítésére való. A FONT2 vékony vonalás karaktereket varázsol a képernyőre, a FONT1 pedig az eredetihez hasonló betűkészletet, persze "ékesebbet". Hogy éppen melyik van kiválasztva, azt a kurzor alakjából tudhatjuk, mert villogó F1 vagy F2 jelenik meg a képernyőn. Az FL és az FS parancs képernyős tontok betöltését és kimentését végzi. A lemezen található E:FONT.BAS BASIC program lehetővé teszi az összes karakter



DDUMP version 2.0 HSOFT 1991.

újratervezését - ha győzzük türelemmel. Ezeket aztán a HPRINT-tel kezelhetjük. A HP <fájlnév> parancs fájlból nyomtat, vagy fájlnév nélkül a képernyő tartalma kerül a printerre. HP\$ való egy sztring kinyomtatására, HPC-vel pedig a teljes karakterkészletet ellenőrizhetjük kinyomtatva. A PFL <fájlnév> parancssal fájlban tárolt printer-karakterkészletet lehet kiküldeni. Megtervezését segíti a PRNFONT.BAS program.

A DDUMP rendszerbővítő szöveges és grafikus video-lapok kinyomtatására készült. Nyolc féle szürkeárnyalat jelenhet meg a kinyomtatott képen, ezeket különböző sűrűségű raszterrel emulálja a mátrixprinter. Tizenhét különböző parancs segíti a

2. STAR LC 24-10-zel, ami persze 24 tús így kissé túlméretezett ENTERPRISE-viszonylatban, mindamellett EPSON kompatibilisnek mondja magát.

3. EPSON LX-400-zal. Tény hogy a kis EPSON volt a leggyesebb, kitűnően felrajzolta az itt látható demó-ábrákat. (A hölgy portréja a lemez DDUMP.DEM nevű, véletlenül sem .BAS kiterjesztésű de mégis BASIC nyelvű programjától származik, miután magáévá tette a KE\$P.SCR fájlt.)

Szóval ügyeskedett az LX-400, de az ékezetes karakterkép nyomtatása, azaz a karakterkészlet betöltés schogy sem sikerült, legyen az HPRINT vagy KP, mindegy. Jó lett volna ha a szerző körülírja, hogy mennyire is legyen az a nyomtató "EPSON kom-

DDUMP DEMO (c) 1992 Hsoft.



DDUMP version 2.0 HSOFT 1991.

megfelelő méretű; palettájú; margójú; fejlécű; keretű stb. nyomtatást.

Az opciókat összefoglalja a DDUMP.STP fájl, amit kedvünk szerint szerkeszthetünk. Megoldható a video-csatorna fájlba küldése, majd később onnét a kinyomtatás. Praktikus hogy a CTRL-P kombinációval is indíthatjuk a printelést, ennek akkor van jelentősége, ha egyébként nem tudnánk parancsot begépelni. A fekete alapon fehér betűket tartalmazó szövegeslap-másolat a DDUMP segítségével készült. Az EXOS bővítéseit tartalmazza német ENTERPRISE-on. A felső négy rendszerbővítés származik a HSOFT-től.

Az ENTERPRISE aktuális karakterkészletével ígér nyomtatást a KP betöltése, ismét EPSON kompatibilis nyomtatót feltételezve. WP-ből F8 után KP-vel, egyébként KP <fájlnév> parancssal indíthatjuk a műveletet. Ez is hasznos szolgáltatásnak látszik, ha "elégé kompatibilis" a printerünk. De erről majd bővebben is szó esik később...

Negyedik opció a VLOAD, ami a video-csatorna betöltését, társa a VSAVE pedig kimentését végzi. Ez funkcióját (és nevét) tekintve megegyezik a német ENTERPRISE gépek beépített parancsaival, ezért csak az "angolosok" számára újdonság.

A lemezen van még egy BORITEK.BAS program, segítségével szabványos boríték címezését oldhatjuk meg gyorsan és látványosan.

Nagy vonalakban ennyit tud a HSOFT új programcsomagja, ami nem kevés, de lássuk mivel marad mégis adós!

Háromféle nyomtatóval is kipróbáltuk:

1. VIDEOTON 21550-nel

patibilis"! Nagyon vagy még jobban?

A különböző nyomtatási kódokat (aláhúzás, dőlt stb.) jól értelmezte a printer.

A HPRINT.DEM nevű szövegfájl kiírása félresikerült az ékezetes karakterek hiánya miatt. A legelső karakterkép-betöltő parancsot nem tudta végrehajtani a nyomtató. Ellenben a 24 tús STAR is szépen kinyomtatta a DDUMP segítségével a videolapok másolatát.

Hiányoljuk még, hogy a német billentyűzetű ENTERPRISE tulajdonosaira nem gondolt a program írója, itt aztán még nehezebb az ékezetes karakterek megtalálása, pedig könnyebb lett volna a betűk logikus elhelyezése.

Kár, hogy az átlagosan néhány kilobájt méretű rendszerbővítő programok egyenként lefoglalnak egy-egy tizenhat kilobájt méretű memória-lapot. Így hamar elfogy a kevéske memóriánk! Áthelyezhető rendszerbővítőként (7-es típus) sokkal kisebb helyen elférnének. (Infó a memóriáról a BASIC-képernyő-másolaton látható.)

A használati utasítás, azaz az összes .INF fájl lehetne egyértelműbb és lenyegre törő. Jó lett volna a teljesen tájékozatlan felhasználóknak és a programozásban jártasoknak külön-külön fejezeteket szentelni.

Összefoglalás: Aki meg akarja vásárolni a PRINT-PACK-ot, jól járjon utána hogy a printerével együtt tud-e dolgozni a program. Ha igen, nem bánja meg a dolgot, és a PRINT-PACK segítségével gyümölcsöztetheti az ENTERPRISE és nyomtatója házasságát.

Bozai Gábor

A győztes játékprogram: CYCLOPS

Induláskor néhány másodperc várakozás után megjelennek az opciók, melyekből kiválaszthatjuk az irányítást. Ha nem csinálunk semmit, akkor kis idő múlva láthatjuk az első 10 játékos nevét és pontszámát.

Indítás az F1-el történik.

A játék célja, hogy a táblán látható golyók közül (meghatározott rendben tőrtendő levételek után) csak egyetlen egy maradjon.

A golyók kiválasztása úgy történik, hogy a mozgatható négyzetel ráálunk a kívánt golyóra, majd megnyomjuk a tüzgombot. Ha később meggondoljuk magunkat, és mégsem ezzel akarunk lépni, akkor a tüzgombot újra megnyomva törölhetjük szándékunkat.)

Az első golyót elegendő kiválasztani, magától eltűnik. Ezután már csak úgy léphetünk fel, le, jobbra vagy balra, hogy átugrunk egy másikat. Ekkor az a golyó amelyiket átléptünk eltűnik. Ha jól csináljuk, akkor a végén csak egy marad, és megjelenik a következő szint. Öt szint van.

Három különböző típusú golyó van, ezek a levételkor fejtik ki hatásukat:

1. SIMA
2. BONUS - több pontot kapunk érte
3. TIME - az idő visszaáll a kiindulási értékre

Ha már nem tudunk többet lépni, vagy elfogy az idő, akkor elveszítjük egyik életünket, amiből kezdetben 3 van.

Ványi Péter

FONTOS! A <<< ... >>> részeket nem kell begepelni!!!

```

1 PROGRAM "CYCLOPS"
2 | Program by Peter Vanyi
3 | (C) REAL PEARL 1992.
4 | ** CYCLOPS **
5 SET 0,4
6 TEXT :CALL CHARACTERS:CLOSE #102
7 SET KEY CLICK OFF
8 SET BORDER 224
9 SET SPEAKER ON
10 SET INTERRUPT STOP OFF
11 SET STATUS OFF:SET 6,1
12 SET TAPE SOUND ON
13 ENVELOPE NUMBER 12;0,63,63,1;0,-63,-63,20;RELEASE;0,-63,-63,200
14 ENVELOPE NUMBER 11;0,63,63,1;0,-63,-63,50
15 ENVELOPE NUMBER 4;0,63,63,1;10,63,63,10
16 ENVELOPE NUMBER 1;0,40,40,1;-50,40,40,3;50,40,40,3
17 ENVELOPE NUMBER 2;0,63,63,1;10,-63,-63,40
18 ENVELOPE NUMBER 3;0,63,63,1;0,-20,-20,20;0,-43,-32,40
19 LET JO=0:LET K=1:LET JY=0
20 NUMERIC KEY(1 TO 5,1 TO 2)
21 FOR I=1 TO 5
22   READ A,B
23   LET KEY(I,2)=A
24   LET KEY(I,1)=B
25 NEXT I
26 DATA 2,79,1,80,8,81,4,65,16,32
27 NUMERIC P(1 TO 2),PS(1 TO 4,1 TO 2),TI
28 FOR I=1 TO 4
29   FOR J=1 TO 2
30     LET PS(I,J)=0
31   NEXT J
32 NEXT I
33 NUMERIC TABLE(-1 TO 9,-1 TO 9),SCORE,HSCORE,BALLS,LEVEL,CODET(1 TO 7,
  0 TO 7),X,Y,LIVE,CUR,CDT(1 TO 7,0 TO 7),NX,NY
34 STRING NAMES(1 TO 10,1 TO 2)*6,LETTERS(1 TO 30)*1
35 FOR I=1 TO 30
36   READ A$
37   LET LETTERS(I)=A$
38 NEXT I
39 DATA A,B,S,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,ö," ",",","!"
<<< a data sorban a Z után szumma karakter, az utolsó előtti karakter ALT-W >>>
40 FOR I=1 TO 10
41   READ A$,B$
42   LET NAMES(I,1)=A$
43   LET NAMES(I,2)=B$
44 NEXT I
45 DATA STEVE,10000,BRUCE,9000,DAVE,8000,NICKO,7000,JANICK,6000,
  MASTER,5000,ICARUS,4000,EDDIE,3000,BATMAN,2000,*?&,1000
46 RESTORE 67
47 FOR I=1 TO 7
48   FOR J=0 TO 7
49     READ A
50     LET CODET(I,J)=A
51   NEXT J
52 NEXT I
53 RESTORE 60
54 FOR I=1 TO 7
55   FOR J=0 TO 7

```

```

56 READ A
57 LET CDT(I,J)=A
58 NEXT J
59 NEXT I
60 DATA 600,0,0,284,364,448,0,0
61 DATA 520,0,200,284,364,448,528,0
62 DATA 440,116,200,284,364,448,528,608
63 DATA 360,116,200,284,364,448,528,608
64 DATA 280,116,200,284,364,448,528,608
65 DATA 200,0,200,284,364,448,528,0
66 DATA 120,0,0,284,364,448,0,0
67 DATA 560,0,0,324,404,488,0,0
68 DATA 480,0,240,324,404,488,568,0
69 DATA 400,156,240,324,404,488,568,648
70 DATA 320,156,240,324,404,488,568,648
71 DATA 240,156,240,324,404,488,568,648
72 DATA 160,0,240,324,404,488,568,0
73 DATA 80,0,0,324,404,488,0,0
74 DEF EXJ(REF CUR,REF TI,REF JY)
75 LET CUR=0
76 SET INK 0:SET 6,1
77 DO
78 PLOT 204+TI,60;204+TI,36,
79 LET TI=TI-4
80 LET JY=JOY(JO)
81 LOOP UNTIL JY<>0 OR TI<=0
82 LET CUR=JY
83 END DEF
84 DEF EXK(REF CUR,REF TI)
85 LET CUR=0
86 SET INK 0:SET 6,1
87 DO
88 PLOT 204+TI,60;204+TI,36,
89 LET TI=TI-3
90 LET IS=INKEY$
91 LOOP UNTIL IS<>" " OR TI<=0
92 IF TI<=0 THEN EXIT DEF
93 FOR I=1 TO 5
94 IF ORD(IS)=KEY(I,1) THEN LET CUR=KEY(I,2):EXIT FOR
95 NEXT I
96 END DEF
97 DEF LETT PICTURE
98 SET 22,1:SET 23,1:SET 24,35:SET 25,2:OPEN #12:"VIDEO:"
99 SET #12:PALETTE 62,254,14,36
100 PLOT #12:0,4;0,68;1116,68;1112,64;4,64;4,8,
101 SET #12:INK 2:PLOT #12:0,0;1116,0;1116,64;1112,60;1112,4;4,4,
102 SET #12:INK 3:PLOT #12:0,48,
103 FOR I=1 TO 30
104 PRINT #12:TAB(I+2);LETTERS(I);
105 NEXT I
106 SET #12:LINE MODE 3
107 END DEF
108 DEF CHAR PICTURE
109 SET 22,1:SET 23,1:SET 24,8:SET 25,2:OPEN #11:"VIDEO:"
110 SET #11:PALETTE 62,254,14,36
111 PLOT #11:0,4;0,68;248,68;244,64;4,64;4,8,
112 SET #11:INK 2
113 PLOT #11:0,0;252,0;252,68;248,64;248,4;4,4,
114 END DEF
115 DEF GAOV PICTURE
116 SET 22,1:SET 23,2:SET 24,18:SET 25,2:OPEN #10:"VIDEO:"
117 SET #10:PALETTE 224,255,2
118 PLOT #10:0,68,:PRINT #10:"GAME OVER":SET #10:INK 2
119 PLOT #10:4,64,:PRINT #10:"GAME OVER"
120 END DEF
121 DEF MODIFY PICTURE
122 SET 22,1:SET 23,1:SET 24,25:SET 25,7:OPEN #8:"VIDEO:"
123 SET #8:PALETTE 62,254,14,36
124 PLOT #8:190,228,:PRINT #8:"MODIFY KEYS"
125 PLOT #8:0,248;792,248;788,244;4,244;4,8;0,4;0,248,
126 SET #8:INK 2
127 PLOT #8:0,0;796,0;796,248;792,244;792,4;4,4,
128 PLOT #8:8,180;788,180,788,172;8,172,788,64;8,64,
129 SET #8:INK 3
130 PLOT #8:68,156,:PRINT #8:"LEFT RIGHT UP DOWN"
131 PLOT #8:160,48,:PRINT #8:"CHOOSER"
132 END DEF
133 DEF SELECT(REF KEY,REF K,REF JO)
134 CALL START SOUND
135 DISPLAY #2:AT 1 FROM 1 TO 17
136 SET #2:PALETTE 146,0,23,74
137 SET FKEY 1 CHR$(49)

```

```

138 SET FKEY 2 CHR$(50)
139 SET FKEY 3 CHR$(51)
140 SET FKEY 4 CHR$(52)
141 SET FKEY 5 CHR$(53)
142 SET #2:INK 2
143 PLOT #2:72,320,:PRINT #2:"F2 KEYBOARD":LET K=1
144 SET #2:INK 1:PLOT #2:72,256,
145 PRINT #2:"F3 INTERNAL"
146 PLOT #2:72,192,:PRINT #2:"F4 EXTERNAL 1"
147 PLOT #2:72,128,:PRINT #2:"F5 MODIFY KEYS"
148 DO
149   WHEN EXCEPTION USE OUTSC
150     SET TIMER 7
151     DO
152       LET I=VAL(INKEY$)
153       LOOP UNTIL I=1 OR I=2 OR I=3 OR I=4 OR I=5
154     END WHEN
155     SET TIMER 0
156     SOUND PITCH 47,DURATION 7,LEFT 100,RIGHT 100
157     SELECT CASE I
158     CASE 2
159       SET #2:INK 2:PLOT #2:72,320,
160       PRINT #2:"F2 KEYBOARD"
161       SET #2:INK 1:PLOT #2:72,256,
162       PRINT #2:"F3 INTERNAL"
163       PLOT #2:72,192,:PRINT #2:"F4 EXTERNAL 1"
164       PLOT #2:72,128,:PRINT #2:"F5 MODIFY KEYS"
165       LET K=1
166     CASE 3
167       SET #2:INK 2:PLOT #2:72,256,
168       PRINT #2:"F3 INTERNAL"
169       SET #2:INK 1:PLOT #2:72,320,
170       PRINT #2:"F2 KEYBOARD"
171       PLOT #2:72,192,:PRINT #2:"F4 EXTERNAL 1"
172       PLOT #2:72,128,:PRINT #2:"F5 MODIFY KEYS"
173       LET JO=0:LET K=0
174     CASE 4
175       SET #2:INK 2:PLOT #2:72,192,
176       PRINT #2:"F4 EXTERNAL 1"
177       SET #2:INK 1:PLOT #2:72,320,
178       PRINT #2:"F2 KEYBOARD"
179       PLOT #2:72,256,:PRINT #2:"F3 INTERNAL"
180       PLOT #2:72,128,:PRINT #2:"F5 MODIFY KEYS"
181       LET JO=1:LET K=0
182     CASE 5
183       SET #2:INK 2:PLOT #2:72,128,
184       PRINT #2:"F5 MODIFY KEYS"
185       SET #2:INK 1:PLOT #2:72,320,
186       PRINT #2:"F2 KEYBOARD"
187       PLOT #2:72,256,:PRINT #2:"F3 INTERNAL"
188       PLOT #2:72,192,:PRINT #2:"F4 EXTERNAL 1"
189       CALL MODIFY_KEYS(KEY)
190     CASE ELSE
191       EXIT DO
192     END SELECT
193   LOOP
194 END DEF
195 DEF INFORMATION PICTURE
196 SET 22,1:SET 23,1:SET 24,35:SET 25,6:OPEN #101:"VIDEO:"
197 SET PALETTE 62,36,254,14
198 PLOT 64,172,:PRINT #101:"LIVES"
199 PLOT 64,136,:PRINT #101:"LEVEL"
200 PLOT 64,100,:PRINT #101:"BALLS"
201 PLOT 64,64,:PRINT #101:"TIME"
202 PLOT 496,136,:PRINT #101:"SCORE"
203 PLOT 400,100,:PRINT #101:"HI-SCORE"
204 SET INK 3
205 PLOT 688,136,:PRINT #101:"00000"
206 PLOT 688,100,:PRINT #101:"10000"
207 PLOT 256,172:PRINT #101:"03"
208 PLOT 256,136:PRINT #101:"01"
209 SET INK 2:PLOT 32,28;0,0;0,212;1116,212;1084,188;32,188;32,28,12,36,PAINT
210 PLOT 892,48;976,176;992,152;991,144;924,48;892,48,916,64,PAINT
211 PLOT 936,48;1000,140;1012,120;960,48;936,48,960,64,PAINT
212 PLOT 972,48;1016,112;1028,92;996,48;972,48,996,64,PAINT
213 PLOT 1008,48;1032,84;1056,48;1008,48,1032,64,PAINT
214 SET INK 3:PLOT 4,0;1116,0;1116,212;1084,184;1084,24;32,24;4,0,36,12,PAINT
215 END DEF
216 DEF MENU PICTURE
217 SET 22,1
218 SET 23,1
219 SET 24,19

```

```

220 SET 25,17
221 OPEN #2:"VIDEO:"
222 SET #2:PALETTE 62,36,254,14
223 PLOT #2:180,560,:PRINT #2:"CYCLOPS"
224 PLOT #2:92,508,:PRINT #2:"SELECT OPTION"
225 PLOT #2:72,384,:PRINT #2:"F1 BEGIN GAME"
226 PLOT #2:72,320,:PRINT #2:"F2 KEYBOARD"
227 PLOT #2:72,256,:PRINT #2:"F3 INTERNAL"
228 PLOT #2:72,192,:PRINT #2:"F4 EXTERNAL 1"
229 PLOT #2:72,128,:PRINT #2:"F5 MODIFY KEYS"
230 PLOT #2:32,444;568,444,568,436;32,436,
231 SET #2:INK 2
232 PLOT #2:0,608;604,608;572,584;28,584;28,32;0,0;0,600,12,64,PAINT
233 SET #2:INK 3
234 PLOT #2:4,0;604,0;604,604;573,580;572,32;28,32;4,0,40,12,PAINT
235 END DEF
236 DEF SCORE PICTURE
237 SET 22,I
238 SET 23,1
239 SET 24,16
240 SET 25,12
241 FOR I=3 TO 7
242 OPEN #I:"VIDEO:"
243 SET #I:PALETTE 224,1,253,YELLOW
244 NEXT I
245 !PUZZLE ONE
246 PLOT #3:264,428;0,0;124,0;320,336;264,428,264,368,PAINT
247 PLOT #3:324,324;136,0;252,0;376,232;324,324,328,232,PAINT
248 PLOT #3:380,224;264,0;372,0;436,128;380,224,372,128,PAINT
249 PLOT #3:440,120;384,0;508,0;440,120,456,20,PAINT
250 !PUZZLE TWO
251 PLOT #4:68,120;128,0;0,0;68,120,68,76,PAINT
252 PLOT #4:72,128;136,0;244,0;128,224;72,128,128,164,PAINT
253 PLOT #4:132,232;256,0;372,0;184,324;132,232,184,276,PAINT
254 PLOT #4:188,336;384,0;508,0;244,428;188,336,244,364,PAINT
255 !PUZZLE THREE
256 PLOT #5:264,428;72,64;396,0;264,428,264,376,PAINT
257 SET #5:INK 0
258 PLOT #5:288,356;144,48,216,36;312,252,353,148;296,20,
259 !PUZZLE FOUR
260 PLOT #6:244,428;436,64;112,0;244,428,244,388,PAINT
261 SET #6:INK 0
262 PLOT #6:220,353;368,44,292,36;192,256,156,148;212,20,
263 !PUZZLE FIVE
264 PLOT #7:228,428;264,428;264,0;228,0;228,428,244,8,PAINT
265 CALL PRNTSC
266 DEF PRNTSC
267 SET 0,4
268 FOR I=3 TO 7
269 SET #I:LINE MODE 3
270 FOR J=3 TO 12
271 SET #I:INK 2
272 PRINT #I,AT J,2:NAME$(J-2,1)
273 PRINT #I,AT J,12,USING "%%%" :VAL(NAME$(J-2,2))
274 NEXT J
275 NEXT I
276 SET 0,20
277 END DEF
278 FOR I=3 TO 7
279 SET #I:INK 3
280 PRINT #I,AT 1,3:"BEST PLAYERS"
281 NEXT I
282 END DEF
283 DEF TURNER
284 RANDOMIZE :LET I=GREEN
285 SET #3:COLOR 1,I
286 DISPLAY #3:AT 6 FROM 1 TO 12
287 CALL WAIT
288 SET #5:COLOR 1,I
289 DISPLAY #5:AT 6 FROM 1 TO 12
290 CALL WAIT
291 LET I=RND(256)
292 SET #7:COLOR 1,I
293 DISPLAY #7:AT 6 FROM 1 TO 12
294 CALL WAIT
295 SET #6:COLOR 1,I
296 DISPLAY #6:AT 6 FROM 1 TO 12
297 CALL WAIT
298 SET #4:COLOR 1,I
299 DISPLAY #4:AT 6 FROM 1 TO 12
300 CALL WAIT
301 SET #5:COLOR 1,I

```

```

302 DISPLAY #5:AT 6 FROM 1 TO 12
303 CALL WAIT
304 SET #7:COLOR 1,I
305 DISPLAY #7:AT 6 FROM 1 TO 12
306 CALL WAIT
307 SET #6:COLOR 1,I
308 DISPLAY #6:AT 6 FROM 1 TO 12
309 CALL WAIT
310 DEF WAIT
311     FOR J=1 TO 40
312     NEXT J
313 END DEF
314 END DEF
315 DEF TABLE PICTURE
316     SET 22,I
317     SET 23,1
318     SET 24,25
319     SET 25,18
320 OPEN #1:"VIDEO:"
321 SET #1:PALETTE 224,31,65,96
322 SET #1:VIDEO MODE 3
323 PLOT #1:276,608;536,608;536,528;616,528;616,448;696,448;696,192;
        616,192;616,112;536,112;536,32;
324 PLOT #1:276,32;276,112;192,112;192,192;108,192;108,448;192,448;
        192,528;276,528;276,608,
325 SET #1:INK 2
326 PLOT #1:284,600;528,600;528,520;608,520;608,440;688,440;688,200;
        608,200;608,120;
327 PLOT #1:528,120;528,40;284,40;284,120;200,120;200,200;116,200;
        116,440;200,440;200,520;284,520;284,600,
328 SET #1:INK 1
329 PLOT #1:280,592,PAINT
330 PLOT #1:248,636;560,636,560,632;248,632,
331 PLOT #1:248,636;76,472,80,472;248,632,
332 PLOT #1:76,472;76,168,80,168;80,472,
333 PLOT #1:80,168;248,8,248,4;76,168,
334 SET #1:INK 3
335 PLOT #1:252,8;560,8,560,4;252,4,
336 PLOT #1:560,4;724,168,720,168;560,8,
337 PLOT #1:720,168;720,472,724,472;724,168,
338 PLOT #1:724,472;564,632,564,628;720,472,
339 SET #1:INK 2
340 PLOT #1:416,620,PAINT
341 PLOT #1:200,200;200,440,284,520;284,120,364,40;364,600,448,600;
        448,40,528,120;528,520,608,440;608,200,
342 PLOT #1:528,120;284,120,200,200;608,200,688,280;116,280,116,360;
        688,360,608,440;200,440,284,520;528,520,
343 END DEF
344 CALL MENU PICTURE
345 CALL TABLE PICTURE
346 CALL INFORMATION PICTURE
347 CALL SCORE PICTURE
348 CALL MODIFY PICTURE
349 CALL GAOV PICTURE
350 CALL LETT_PICTURE
351 CALL CHAR_PICTURE
352 CALL GLORY_PICTURE
353 SET 0,20
354 DO
355     CALL SELECT(KEY,K,JO)
356     SET #2:PALETTE 224,224,224,224
357     CALL TIMEUP(TI,650)
358     DISPLAY #1:AT 1 FROM 1 TO 18
359     SET #1:PALETTE 224,31,65,96
360     DISPLAY #101:AT 20 FROM 1 TO 6
361     SET PALETTE 62,36,254,14
362     LET LEVEL=1:LET SCORE=0:LET LIVE=3
363     SET INK 0:PLOT 256,172,:PRINT #101:"ÜÜÜ" <<< 3*ALT-N >>>
364     SET INK 3:PLOT 256,172,:PRINT #101,USING "%%":LIVE
365     SET INK 0:PLOT 688,100,:PRINT #101:"ÜÜÜÜÜ" <<< 5*ALT-N >>>
366     SET INK 3:PLOT 688,100,:PRINT #101,USING "%%%%" :VAL(NAME$(1,2))
367     CALL SC(SCORE,0)
368     CALL FULL(BALLS,TABLE,LEVEL)
369     LET X,Y,NX,NY=4:CALL PNT(X,Y):LET P(1),P(2)=0:LET FRS=0
370     DO
371     DO
372         IF K=1 THEN CALL EXK(CUR,TI)
373         IF K=0 THEN CALL EXJ(CUR,TI,JY)
374     LOOP UNTIL CUR<>0 OR TI<=0
375     IF TI<=0 THEN
376         CALL TIME_OVER
377         WAIT 5

```



```

378 CALL DEATH(LIVE,FRS,P)
379 LET CUR=0
380 END IF
381 SELECT CASE CUR
382 CASE 1
383 CALL STEP R(NX,NY)
384 CALL PNT(X,Y):CALL PNT(NX,NY)
385 CASE 2
386 CALL STEP L(NX,NY)
387 CALL PNT(X,Y):CALL PNT(NX,NY)
388 CASE 4
389 CALL STEP D(NX,NY)
390 CALL PNT(X,Y):CALL PNT(NX,NY)
391 CASE 8
392 CALL STEP U(NX,NY)
393 CALL PNT(X,Y):CALL PNT(NX,NY)
394 CASE 16
395 IF TABLE(NX,NY)<>0 THEN CALL CHOOSE(TABLE,FRS,P,PS,NX,NY)
396 IF TABLE(NX,NY)=0 THEN CALL MOVE(P,PS,TABLE)
397 CASE ELSE
398 END SELECT
399 LET X=NX:LET Y=NY
400 LOOP UNTIL LEVEL=6 OR LIVE=0
401 CALL GAMOV
402 IF SCORE>=VAL(NAMES(10,2)) THEN CALL BEST(NAMES)
403 IF SCORE<VAL(NAMES(10,2)) THEN WAIT 2
404 SET PALETTE 224,224,224,224
405 LOOP
406 DEF CHARACTERS
407 SET CHARACTER 145,60,126,207,207,223,255,255,126,60
408 SET CHARACTER 151,0,102,255,255,255,126,60,24,0
409 SET CHARACTER 65,0,62,114,114,114,126,114,114,0:
410 SET CHARACTER 66,0,124,114,114,124,114,114,124,0:
411 SET CHARACTER 67,0,30,58,112,112,112,58,30,0
412 SET CHARACTER 68,0,56,116,114,114,114,114,124,0:
413 SET CHARACTER 69,0,62,114,112,124,112,114,62,0:
414 SET CHARACTER 70,0,62,112,112,124,112,112,48
415 SET CHARACTER 71,0,62,114,112,112,118,114,62,0:
416 SET CHARACTER 72,0,114,114,114,126,114,114,114,0:
417 SET CHARACTER 73,0,124,56,56,56,56,56,124,0
418 SET CHARACTER 74,0,14,14,14,78,78,78,60,0:
419 SET CHARACTER 75,0,114,114,116,124,116,114,114,0:
420 SET CHARACTER 76,0,48,112,112,112,112,112,62,0
421 SET CHARACTER 77,0,118,106,106,106,98,98,98,0:
422 SET CHARACTER 78,0,114,114,122,118,114,114,114,0:
423 SET CHARACTER 79,0,60,114,114,114,114,114,60,0
424 SET CHARACTER 80,0,60,114,114,114,124,112,112,0:
425 SET CHARACTER 82,0,60,114,114,114,124,116,118,0:
426 SET CHARACTER 81,0,60,114,114,114,114,118,62,3
427 SET CHARACTER 83,0,60,114,120,28,78,78,60,0:
428 SET CHARACTER 84,0,254,56,56,56,56,56,56,0:
429 SET CHARACTER 85,0,114,114,114,114,114,118,62,0
430 SET CHARACTER 86,0,114,114,114,114,114,116,56,0:
431 SET CHARACTER 87,0,113,113,117,117,117,127,123,0:
432 SET CHARACTER 88,0,71,71,46,28,28,60,114,0
433 SET CHARACTER 89,0,99,99,54,28,28,28,28,0:
434 SET CHARACTER 90,0,126,14,30,60,120,112,126,0:
435 SET CHARACTER 33,0,24,24,24,24,24,0,24,0
436 SET CHARACTER 63,0,60,102,14,28,24,0,24,0:
437 SET CHARACTER 58,0,0,0,48,0,0,48,0,0
438 SET CHARACTER 48,0,60,114,118,122,114,114,60,0:
439 SET CHARACTER 49,0,60,124,28,28,28,28,28,0:
440 SET CHARACTER 50,0,60,110,14,28,56,56,62,0
441 SET CHARACTER 51,0,60,78,14,62,14,78,60,0:
442 SET CHARACTER 52,0,48,112,116,116,126,4,4:
443 SET CHARACTER 53,0,62,112,112,124,14,14,124,0
444 SET CHARACTER 54,0,30,48,96,124,114,98,126,0:
445 SET CHARACTER 55,0,126,14,28,56,56,56,56,0:
446 SET CHARACTER 56,0,60,36,60,114,114,114,60,0
447 SET CHARACTER 57,0,62,50,50,30,6,12,56,0
448 END DEF
449 DEF MODIFY_KEYS(REF KEY)
450 DO
451 SET #8:PALETTE 62,254,14,36
452 FOR I=1 TO 5
453 LET KEY(I,1)=0
454 NEXT I
455 DISPLAY #8:AT 19 FROM 1 TO 7
456 SET #8:INK 2
457 PLOT #8:68,156,:PRINT #8:"LEFT"
458 DEF ASKKEY(REF KEY,A)
459 DO

```

```

435 DO
436 LET I$=INKEY$
437 LOOP UNTIL I$<>"
438 LET I=ORD(UCASE$(I$))
439 IF A=2 AND I=KEY(1,1) THEN LET I=0
440 IF A=3 AND I=KEY(1,1) OR I=KEY(2,1) THEN LET I=0
441 IF A=4 AND I=KEY(1,1) OR I=KEY(2,1) OR I=KEY(3,1) THEN LET I=0
442 IF A=5 AND I=KEY(1,1) OR I=KEY(2,1) OR I=KEY(3,1) OR I=KEY(4,1)
    THEN LET I=0
443 IF A=5 AND I=32 THEN EXIT DO
444 LOOP UNTIL I>=48 AND I<=57 OR I>=65 AND I<=90
445 SOUND PITCH 58,DURATION 7,LEFT 100,RIGHT 100
446 LET KEY(A,1)=I
447 LET ASKKEY=I
448 END DEF
449 PLOT #8:116,108,:PRINT #8:CHR$(ASKKEY(KEY,1))
450 SET #8:INK 3:PLOT #8:68,156,
451 PRINT #8:"LEFT"
452 PLOT #8:68,156,:SET #8:INK 2
453 PRINT #8:" RIGHT"
454 PLOT #8:324,108,:PRINT #8:CHR$(ASKKEY(KEY,2))
455 SET #8:INK 3:PLOT #8:68,156,
456 PRINT #8:" RIGHT"
457 PLOT #8:68,156,:SET #8:INK 2
458 PRINT #8:" UP"
459 PLOT #8:504,108,:PRINT #8:CHR$(ASKKEY(KEY,3))
460 SET #8:INK 3:PLOT #8:68,156,
461 PRINT #8:" UP"
462 PLOT #8:68,156,:SET #8:INK 2
463 PRINT #8:" DOWN"
464 PLOT #8:664,108,:PRINT #8:CHR$(ASKKEY(KEY,4))
465 SET #8:INK 3:PLOT #8:68,156,
466 PRINT #8:" DOWN"
467 PLOT #8:160,48,:SET #8:INK 2
468 PRINT #8:"CHOOSER"
469 PLOT #8:452,48,:PRINT #8:CHR$(ASKKEY(KEY,5))
470 SET #8:INK 3:PLOT #8:160,48,
471 PRINT #8:"CHOOSER"
472 SET #8:INK 2:PLOT #8:216,228,:PRINT #8:"
473 DO
474 LET I$=INKEY$
475 SET #8:INK 2:PLOT #8:216,228,:PRINT #8:"
476 SET #8:INK 0:PLOT #8:216,228,:PRINT #8:"
477 LOOP UNTIL I$="Y" OR I$="N" OR I$="y" OR I$="n"
478 SET #8:PALETTE 224,224,224,224
479 SET #8:INK 0:PLOT #8:216,228,:PRINT #8:"
480 PLOT #8:116,108,:PRINT #8:CHR$(KEY(1,1))
481 PLOT #8:324,108,:PRINT #8:CHR$(KEY(2,1))
482 PLOT #8:504,108,:PRINT #8:CHR$(KEY(3,1))
483 PLOT #8:664,108,:PRINT #8:CHR$(KEY(4,1))
484 PLOT #8:452,48,:PRINT #8:CHR$(KEY(5,1))
485 LOOP UNTIL I$="Y" OR I$="y"
486 SET #2:INK 2:PLOT #2:72,320,
487 PRINT #2:"F2 KEYBOARD"
488 LET K=1
489 SET #2:INK 1
490 PLOT #2:72,128,:PRINT #2:"F5 MODIFY KEYS"
491 END DEF
492 DEF TORF(A,B)
493 IF TABLE(A,B)=0 THEN LET TORF=0:EXIT DEF
494 IF TABLE(A+1,B)<>0 AND TABLE(A+1,B)<>2 AND TABLE(A+2,B)=0 THEN
    LET TORF=1:EXIT DEF
495 IF TABLE(A,B-1)<>0 AND TABLE(A,B-1)<>2 AND TABLE(A,B-2)=0 THEN
    LET TORF=1:EXIT DEF
496 IF TABLE(A-1,B)<>0 AND TABLE(A-1,B)<>2 AND TABLE(A-2,B)=0 THEN
    LET TORF=1:EXIT DEF
497 IF TABLE(A,B+1)<>0 AND TABLE(A,B+1)<>2 AND TABLE(A,B+2)=0 THEN
    LET TORF=1:EXIT DEF
498 LET TORF=0
499 END DEF
500 DEF FULL(REF BALLS,REF TABLE,L)
501 CLEAR SOUND
502 SET 0,4
503 SELECT CASE L
504 CASE 1
505 RESTORE 537
506 LET BALLS=9
507 CASE 2
508 RESTORE 549
509 LET BALLS=14
510 CASE 3
511 RESTORE 561

```

OK Y/N"

OK Y/N"

OK Y/N"

OK Y/N"

```
512 LET BALLS=18
513 CASE 4
514 RESTORE 573
515 LET BALLS=26
516 CASE ELSE
517 RESTORE 585
518 LET BALLS=38
519 END SELECT
520 CALL COUB(BALLS)
521 FOR I=-1 TO 9
522 FOR J=-1 TO 9
523 READ X
524 LET TABLE(I,J)=X
525 SELECT CASE X
526 CASE 1
527 CALL BALL(I,J,1)
528 CASE 3,4
529 CALL BALL(I,J,2)
530 CASE ELSE
531 END SELECT
532 NEXT J
533 NEXT I
534 SET 0,20
535 END DEF
536 !LEVEL 1
537 DATA 2,2,2,2,2,2,2,2,2,2,2,2
538 DATA 2,2,2,2,2,2,2,2,2,2,2,2
539 DATA 2,2,2,2,0,0,0,0,2,2,2,2
540 DATA 2,2,2,0,0,0,0,0,2,2,2,2
541 DATA 2,2,0,0,0,3,3,0,0,2,2,2
542 DATA 2,2,0,0,3,3,3,3,0,2,2,2
543 DATA 2,2,0,0,0,3,3,0,0,2,2,2
544 DATA 2,2,2,0,0,0,0,0,2,2,2,2
545 DATA 2,2,2,2,0,0,0,0,2,2,2,2
546 DATA 2,2,2,2,2,2,2,2,2,2,2,2
547 DATA 2,2,2,2,2,2,2,2,2,2,2,2
548 !LEVEL 2
549 DATA 2,2,2,2,2,2,2,2,2,2,2,2
550 DATA 2,2,2,2,2,2,2,2,2,2,2,2
551 DATA 2,2,2,2,0,0,0,0,2,2,2,2
552 DATA 2,2,2,0,0,1,0,0,2,2,2,2
553 DATA 2,2,0,0,1,3,1,0,0,2,2,2
554 DATA 2,2,0,1,4,1,4,1,0,2,2,2
555 DATA 2,2,0,0,1,3,1,0,0,2,2,2
556 DATA 2,2,2,0,0,1,0,0,2,2,2,2
557 DATA 2,2,2,2,0,0,0,0,2,2,2,2
558 DATA 2,2,2,2,2,2,2,2,2,2,2,2
559 DATA 2,2,2,2,2,2,2,2,2,2,2,2
560 !LEVEL 3
561 DATA 2,2,2,2,2,2,2,2,2,2,2,2
562 DATA 2,2,2,2,2,2,2,2,2,2,2,2
563 DATA 2,2,2,2,0,0,0,0,2,2,2,2
564 DATA 2,2,2,0,0,0,0,0,2,2,2,2
565 DATA 2,2,0,1,1,3,1,1,0,2,2,2
566 DATA 2,2,1,3,1,1,1,3,1,2,2,2
567 DATA 2,2,0,1,1,3,1,1,0,2,2,2
568 DATA 2,2,2,0,0,0,0,0,2,2,2,2
569 DATA 2,2,2,2,0,0,0,0,2,2,2,2
570 DATA 2,2,2,2,2,2,2,2,2,2,2,2
571 DATA 2,2,2,2,2,2,2,2,2,2,2,2
572 !LEVEL 4
573 DATA 2,2,2,2,2,2,2,2,2,2,2,2
574 DATA 2,2,2,2,2,2,2,2,2,2,2,2
575 DATA 2,2,2,2,0,1,0,2,2,2,2,2
576 DATA 2,2,2,0,1,4,1,0,2,2,2,2
577 DATA 2,2,0,1,3,1,3,1,0,2,2,2
578 DATA 2,2,1,4,1,3,1,4,1,2,2,2
579 DATA 2,2,0,1,3,1,3,1,0,2,2,2
580 DATA 2,2,2,0,1,4,1,0,2,2,2,2
581 DATA 2,2,2,2,0,1,0,2,2,2,2,2
582 DATA 2,2,2,2,2,2,2,2,2,2,2,2
583 DATA 2,2,2,2,2,2,2,2,2,2,2,2
584 !LEVEL 5
585 DATA 2,2,2,2,2,2,2,2,2,2,2,2
586 DATA 2,2,2,2,2,2,2,2,2,2,2,2
587 DATA 2,2,2,2,1,4,1,2,2,2,2,2
588 DATA 2,2,2,1,3,1,3,1,2,2,2,2
589 DATA 2,2,1,3,1,1,1,3,1,2,2,2
590 DATA 2,2,4,1,1,3,1,1,4,2,2,2
591 DATA 2,2,1,3,1,1,1,3,1,2,2,2
592 DATA 2,2,2,1,3,1,3,1,2,2,2,2
```

```

593 DATA 2,2,2,2,1,4,1,2,2,2,2
594 DATA 2,2,2,2,2,2,2,2,2,2,2
595 DATA 2,2,2,2,2,2,2,2,2,2,2
596 DEF ANTBALL(A,B)
597   IF TABLE(A,B)<>1 THEN CALL KSN
598   IF TABLE(A,B)=1 THEN CALL SND
599   SET #1:LINE MODE 0
600   SET #1:INK 1
601   PLOT #1:CODET(A,B),CODET(A,0),ELLIPSE 35,35,PAINT
602   SET #1:INK 0
603   PLOT #1:CODET(A,B),CODET(A,0),ELLIPSE 35,35,PAINT
604 END DEF
605 DEF PNT(A,B)
606   SET #1:INK 1
607   SET #1:LINE MODE 3
608   PLOT #1:CDT(A,B),CDT(A,0);CDT(A,B)+83,CDT(A,0);CDT(A,B)+81,CDT(A,0)-83;
     CDT(A,B),CDT(A,0)-83;CDT(A,B),CDT(A,0),
609 END DEF
610 DEF STEP R(REF NX,REF NY)
611   LET NX=X
612   LET NY=Y+1
613   IF Y=7 THEN LET NY=NY-1
614   IF TABLE(NX,NY)=2 THEN LET NY=NY-1
615 END DEF
616 DEF STEP L(REF NX,REF NY)
617   LET NX=X
618   LET NY=Y-1
619   IF Y=1 THEN LET NY=NY+1
620   IF TABLE(NX,NY)=2 THEN LET NY=NY+1
621 END DEF
622 DEF STEP U(REF NX,REF NY)
623   LET NX=X-1
624   LET NY=Y
625   IF X=1 THEN LET NX=NX+1
626   IF TABLE(NX,NY)=2 THEN LET NX=NX+1
627 END DEF
628 DEF STEP D(REF NX,REF NY)
629   LET NY=Y
630   LET NX=X+1
631   IF X=7 THEN LET NX=NX-1
632   IF TABLE(NX,NY)=2 THEN LET NX=NX-1
633 END DEF
634 !
635 DEF BOR(A,B)
636   SET #1:LINE MODE 3
637   SET #1:INK 1
638   PLOT #1:CODET(A,B),CODET(A,0),ELLIPSE 35,35,
639 END DEF
640 DEF CHOOSE(REF TABLE,REF FRs,REF P,REF PS,A,B)
641   IF FRs=0 THEN LET TABLE(A,B)=0:CALL BOR(A,B):CALL ANTBALL(A,B):
     LET FRs=1:CALL COUB(BALLS):CALL SC(SCORE,50):EXIT DEF
642   IF A=P(1) AND B=P(2) THEN LET P(1),P(2)=0:CALL BOR(A,B):EXIT DEF
643   IF P(1)<>0 THEN EXIT DEF
644   FOR I=1 TO 4
645     FOR J=1 TO 2
646       LET PS(I,J)=0
647     NEXT J
648   NEXT I
649   IF TORF(A,B)=0 THEN
650     SOUND PITCH 15,DURATION 3
651     EXIT DEF
652   END IF
653   CALL CHOOSE SOUND
654   CALL BOR(A,B)
655   LET P(1)=A:LET P(2)=B
656   CALL EXA(PS,A,B)
657 END DEF
658 DEF EXA(REF PS,F,G)
659   IF TABLE(F+1,G)<>0 AND TABLE(F+1,G)<>2 AND TABLE(F+2,G)=0 THEN
     LET PS(1,1)=F+2:LET PS(1,2)=G
660   IF TABLE(F,G-1)<>0 AND TABLE(F,G-1)<>2 AND TABLE(F,G-2)=0 THEN
     LET PS(2,1)=F:LET PS(2,2)=G-2
661   IF TABLE(F-1,G)<>0 AND TABLE(F-1,G)<>2 AND TABLE(F-2,G)=0 THEN
     LET PS(3,1)=F-2:LET PS(3,2)=G
662   IF TABLE(F,G+1)<>0 AND TABLE(F,G+1)<>2 AND TABLE(F,G+2)=0 THEN
     LET PS(4,1)=F:LET PS(4,2)=G+2
663 END DEF
664 END DEF
665 DEF MOVE(REF P,REF PS,REF TABLE)
666   IF P(1)=0 THEN EXIT DEF
667   LET Q=0:LET S=1
668   FOR I=1 TO 4

```

```

669     IF NX=PS(I,1) AND NY=PS(I,2) THEN LET Q=I:EXIT FOR
670     NEXT I
671     IF Q=0 THEN
672         SOUND PITCH 15,DURATION 3
673         EXIT DEF
674     END IF
675     CALL CHOOSE_SOUND
676     SELECT CASE Q
677     CASE 1
678         IF TABLE(P(1)+1,P(2))=3 THEN CALL SC(SCORE,100)
679         IF TABLE(P(1)+1,P(2))=4 THEN CALL TIMEUP(TI,650):CALL SC(SCORE,50)
680         CALL ANTBALL(P(1),P(2)):CALL ANTBALL(P(1)+1,P(2))
681         IF TABLE(P(1),P(2))=1 THEN CALL BALL(NX,NY,1)
682         IF TABLE(P(1),P(2))=4 OR TABLE(P(1),P(2))=3 THEN CALL BALL(NX,NY,2)
683         LET TABLE(NX,NY)=TABLE(P(1),P(2)):LET TABLE(P(1),P(2)),TABLE(P(1)+1,
        P(2))=0
684     CALL SC(SCORE,50)
685     CASE 2
686         IF TABLE(P(1),P(2)-1)=3 THEN CALL SC(SCORE,100)
687         IF TABLE(P(1),P(2)-1)=4 THEN CALL TIMEUP(TI,650):CALL SC(SCORE,50)
688         CALL ANTBALL(P(1),P(2)):CALL ANTBALL(P(1),P(2)-1)
689         IF TABLE(P(1),P(2))=1 THEN CALL BALL(NX,NY,1)
690         IF TABLE(P(1),P(2))=4 OR TABLE(P(1),P(2))=3 THEN CALL BALL(NX,NY,2)
691         LET TABLE(NX,NY)=TABLE(P(1),P(2)):LET TABLE(P(1),P(2)),TABLE(P(1),
        P(2)-1)=0
692     CALL SC(SCORE,50)
693     CASE 3
694         IF TABLE(P(1)-1,P(2))=3 THEN CALL SC(SCORE,100)
695         IF TABLE(P(1)-1,P(2))=4 THEN CALL TIMEUP(TI,650):CALL SC(SCORE,50)
696         CALL ANTBALL(P(1),P(2)):CALL ANTBALL(P(1)-1,P(2))
697         IF TABLE(P(1),P(2))=1 THEN CALL BALL(NX,NY,1)
698         IF TABLE(P(1),P(2))=4 OR TABLE(P(1),P(2))=3 THEN CALL BALL(NX,NY,2)
699         CALL SC(SCORE,50)
700         LET TABLE(NX,NY)=TABLE(P(1),P(2)):LET TABLE(P(1),P(2)),TABLE(P(1)-1,
        P(2))=0
701     CASE 4
702         IF TABLE(P(1),P(2)+1)=3 THEN CALL SC(SCORE,100)
703         IF TABLE(P(1),P(2)+1)=4 THEN CALL TIMEUP(TI,650):CALL SC(SCORE,50)
704         CALL ANTBALL(P(1),P(2)):CALL ANTBALL(P(1),P(2)+1)
705         IF TABLE(P(1),P(2))=1 THEN CALL BALL(NX,NY,1)
706         IF TABLE(P(1),P(2))=4 OR TABLE(P(1),P(2))=3 THEN CALL BALL(NX,NY,2)
707         LET TABLE(NX,NY)=TABLE(P(1),P(2)):LET TABLE(P(1),P(2)),TABLE(P(1),
        P(2)+1)=0
708     CALL SC(SCORE,50)
709     END SELECT
710     CALL COUB(BALLS)
711     LET P(1),P(2)=0
712     FOR I=1 TO 4
713         FOR J=1 TO 2
714             LET PS(I,J)=0
715         NEXT J
716     NEXT I
717     IF LEVEL=5 AND BALLS=37 THEN EXIT DEF
718     IF LEVEL=6 THEN EXIT DEF
719     FOR I=1 TO 7
720         FOR J=1 TO 7
721             LET S=0
722             IF TABLE(I,J)<>2 AND TABLE(I,J)<>0 AND TORF(I,J) THEN LET S=1:
                EXIT FOR
723         NEXT J
724         IF S=1 THEN EXIT FOR
725     NEXT I
726     IF S=0 THEN CALL DEATH(LIVE,FRS,P)
727     END DEF
728     DEF BALL(A,B,C)
729     SET #1:LINE MODE 0
730     SELECT CASE C
731     CASE 1
732         SET #1:INK 3
733         PLOT #1:CODET(A,B),CODET(A,0),ELLIPSE 31,31,PAINT
734         SET #1:INK 1
735         PLOT #1:CODET(A,B)+14,CODET(A,0)+10,ELLIPSE 3,4,PAINT
736     CASE ELSE
737         SET #1:INK 2
738         PLOT #1:CODET(A,B),CODET(A,0),ELLIPSE 31,31,PAINT
739         SET #1:INK 1
740         PLOT #1:CODET(A,B)+14,CODET(A,0)+10,ELLIPSE 3,4,PAINT
741     END SELECT
742     END DEF
743     DEF TIMEUP(REF TI,A)
744     SET INK 3
745     PLOT 204,60;204+A,60;204+A,36;204,36;204,60,204+A-4,40,PAINT

```

```

746 LET TI=A
747 END DEF
748 DEF COUB(REF BALLS)
749 LET BALLS=BALLS-1
750 SET INK 0:PLOT 256,100,:PRINT #101:"ÜÜ" <<< 2*ALT-N >>>
751 SET INK 3:PLOT 256,100,:PRINT #101,USING "%%":BALLS
752 IF BALLS=1 THEN CALL LEVELUP(LEVEL,FRS,P)
753 END DEF
754 DEF SC(REF SCORE,A)
755 LET SCORE=SCORE+A
756 SET INK 0:PLOT 688,136,:PRINT #101:"ÜÜÜÜ" <<< 5*ALT-N >>>
757 SET INK 3:PLOT 688,136,:PRINT #101,USING "%%%%":SCORE
758 END DEF
759 HANDLER OUTSC
760 LET B=0
761 SET TIMER 7
762 SET #2:PALETTE 224,224,224,224
763 WHEN EXCEPTION USE BACK
764 DO
765 CALL TURNER
766 LOOP UNTIL B=1 OR INKEY$<>""
767 END WHEN
768 SET TIMER 7
769 DISPLAY #2:AT 1 FROM 1 TO 17
770 CALL START SOUND
771 SET #2:PALETTE 62,36,254,14
772 RETRY
773 END HANDLER
774 DEF DEATH(REF LIVE,REF FRS,REF P)
775 LET LIVE=LIVE-1
776 IF LIVE=0 THEN EXIT DEF
777 SET INK 0:PLOT 256,172,:PRINT #101:"ÜÜÜ" <<< 4*ALT-N >>>
778 SET INK 3:PLOT 256,172,:PRINT #101,USING "%%":LIVE
779 CALL CLTBL:CALL FULL(BALLS,TABLE,LEVEL):CALL TIMEUP(TI,650):LET FRS=0:
LET P(1),P(2)=0
780 END DEF
781 DEF CLTBL
782 FOR I=1 TO 7
783 FOR J=1 TO 7
784 IF TABLE(I,J)<>2 AND TABLE(I,J)<>0 THEN CALL ANTBALL(I,J)
785 NEXT J
786 NEXT I
787 END DEF
788 DEF LEVELUP(REF LEVEL,REF FRS,REF P)
789 LET LEVEL=LEVEL+1
790 CALL SC(SCORE,500)
791 IF LEVEL=6 THEN EXIT DEF
792 SET INK 0:PLOT 256,136,:PRINT #101:"ÜÜ" <<< 3*ALT-N >>>
793 SET INK 3:PLOT 256,136,:PRINT #101,USING "%%":LEVEL
794 CALL CLTBL:CALL FULL(BALLS,TABLE,LEVEL):LET FRS=0:CALL TIMEUP(TI,650):
LET P(1),P(2)=0
795 END DEF
796 DEF CHOOSE SOUND
797 CLEAR SOUND
798 SOUND PITCH 83,DURATION 15,ENVELOPE 1,SOURCE 2
799 END DEF
800 DEF START SOUND
801 SOUND PITCH 84,DURATION 40,ENVELOPE 2,SYNC 1,STYLE 128
802 SOUND PITCH 88,DURATION 40,ENVELOPE 2,SYNC 1,SOURCE 2,STYLE 128
803 END DEF
804 DEF TIME OVER
805 CLEAR SOUND
806 FOR I=1 TO 3
807 SOUND PITCH 25,ENVELOPE 3,SYNC 1,DURATION 60
808 SOUND PITCH 17,ENVELOPE 3,SYNC 1,SOURCE 1,DURATION 60
809 NEXT I
810 END DEF
811 DEF GAMOV
812 CALL CLTBL:CALL PNT(X,Y)
813 CLEAR SOUND
814 SET #1:PALETTE 224,224,224,224
815 SET PALETTE 224,224,224,224
816 IF LEVEL=6 THEN
817 CALL START SOUND
818 DISPLAY #13:AT 2 FROM 1 TO 6
819 SET #13:PALETTE 2,255,0,65
820 DO
821 LOOP UNTIL INKEY$<>""
822 SET #13:PALETTE 224,224,224,224
823 END IF
824 DISPLAY #10:AT 10 FROM 1 TO 2
825 CALL TOTOTU

```

```

826 WAIT 4
827 END DEF
828 HANDLER BACK
829 LET B=1
830 KEYV
831 END HANDLER
832 DEF REN(REF NAMES)
833 SET 0,4
834 LET M$=""
835 FOR I=1 TO 10
836   FOR J=1 TO 10
837     IF VAL(NAMES{I,2})<VAL(NAMES{J,2}) THEN
838       LET M$=NAMES{I,1};LET NAMES{I,1}=NAMES{J,1};LET NAMES{J,1}=M$
839       LET M$=NAMES{I,2};LET NAMES{I,2}=NAMES{J,2};LET NAMES{J,2}=M$
840     END IF
841   NEXT J
842 NEXT I
843 SET 0,20
844 END DEF
845 DEF REST(REF NAMES)
846 CALL PRNTRC
847 LET NAMES(10,2)=STR$(SCORE)
848 LET NAMES(10,1)=RANS(LETTERS,NAMES,CUR)
849 CALL REN(NAMES)
850 CALL PRNTRC
851 END DEF
852 DEF NAINS(REF LETTERS,REF NAMES,REF CUP
853 LET S=16;LET MNS="" ;SEY #11;INK 1;LET U=0;LET LETTERS(30)="
854 DISPLAY #12:AT 14 FROM 1 TO 2
855 DISPLAY #11:AT 17 FROM 1 TO 2
856 SET #12:PALETTE 62,254,14,36
857 SET #11:PALETTE 62,254,14,36
858 SET #12:INK 1;PLOT #12:0,48,;PRINT #12:TAB$(LETTERS(S-2)
859 DO
860 LET CUR=0
861 PLOT #12:0,48,
862 DO
863 LET CUR=JOY{JO}
864 LOOP UNTIL CUR=1 OR CUR=2 OR CUR=16
865 SELECT CASE CUR
866 CASE 1
867 PRINT #12:TAB$(S);LETTERS(S-2)
868 IF S=32 THEN LET S=31
869 LET S=S+1
870 PRINT #12:TAB$(S);LETTERS(S-2)
871 CASE 2
872 PRINT #12:TAB$(S);LETTERS(S-2)
873 IF S=3 THEN LET S=4
874 LET S=S-1
875 PRINT #12:TAB$(S);LETTERS(S-2)
876 CASE ELSE
877 SOUND PITCH 58,DURATION 7,LEFT 100,RIGHT 100
878 IF S=32 THEN LET U=1
879 LET MNS=MNS&LETTERS(S-2)
880 PLOT #11:32,48,;PRINT #11:MNS
881 FOR G=1 TO 40
882 NEXT G
883 END SELECT
884 LOOP UNTIL LEN(MNS)=6 OR U=1
885 SET #12:PALETTE 224,224,224,224
886 SET #11:PALETTE 224,224,224,224
887 PRINT #12:TAB$(S);LETTERS(S-2)
888 SET #11:INK 0;PLOT #11:32,48,;PRINT #11:"000000" <<< 6*ALT-N >>>
889 LET NAINS=MNS
890 END DEF
891 DEF SND
892 SOUND DURATION 2,SOURCE 3
893 END DEF
894 DEF KSN
895 CLEAR QUEUE 1
896 SOUND SOURCE 3,DURATION 3,SYNC 1
897 SOUND PITCH 79,DURATION 100,ENVELOPE 11,SYNC 1,SOURCE 1
898 END DEF
899 DEF TOTOTU
900 FOR I=1 TO 4
901 READ R
902 SOUND PITCH R,DURATION 10,ENVELOPE 12,SYNC 1
903 SOUND PITCH R+7,DURATION 10,SOURCE 1,SYNC 1,ENVELOPE 12
904 NEXT I
905 DATA 26,26,26,24
906 END DEF
907 DEF GLORY PICTURE
908 SET 22,1;SET 23,2;SET 24,40;SET 25,6;OPEN #13:"VIDEO:"
909 SET #13:PALETTE 2,255,0,GREEN
910 PLOT #13:0,0,0,215;1279,215,
911 SET #13:INK 1;PLOT #13:8,0;1279,0;1279,210,
912 SET #13:INK 2
913 PLOT #13:180,190,;PRINT #13:"CONGRATULATION"
914 PLOT #13:110,90,
915 PRINT #13:"YOU SAVED THEM..."
916 SET #13:INK 3
917 PLOT #13:185,195,;PRINT #13:"CONGRATULATION"
918 PLOT #13:115,95,
919 PRINT #13:"YOU SAVED THEM..."
920 END DEF

```

MEGRENDELŐLAP

Megrendelem az ENTERPRESS című lapot

..... példányban az alábbi címmel:

Megrendelő neve:

.....

Címe:

.....
(város, körzet)

.....
(utca, tér, lakótelep)

.....
(emelt, a jó)

.....
lényviszám:

Előfizetési díj egy évre 294 Ft, félévre 147 Ft.

Az előfizetési díjat a részleges költéselőiról a postautalványon egyenlően ki.

.....
aláírás

A megrendelőlapot barátságosan bérmentesítve az alábbi címre kérjük feladni:

Hírlapelőfizetési és Lapelátási Iroda
Budapest
1905

mikrovilág

Az ENTERPRESS előző számai korlátozott példányszámban még megrendelhetők a volt kiadó címen (MATRIX Kft. 8000 Székesfehérvár, Zichy liget 10.), vagy megvásárolhatók a Műszaki Könyvtárházban (Bp. VI. ker. Lizi F. tér 9.).

Tisztelt Olvasóink!

Arra kérjük Önöket, hogy önreklameiket ne a szerkesztőség, hanem a volt kiadó (Matrix Kft.) címére küldjék, mert így sokkal gyorsabban juthatnak hozzá kedvenc lapjukhoz. Köszönjük!

A szerkesztőség

Apróhirdetések

EP turbósítás 6MHz-re, betöltő memóriabővítés 320K-ra!
Külön-külön 3300 Ft, együtt pedig csak 5800 Ft!!!
Bosai Gábor 8000 Székesfehérvár,
József A. u. 62/A 1/2. Tel.: 22/310-665.

Feladó ENTERPRISE számítógép, tartozékok: 3.5" (+ saját kérésre 5.25", 360 KB), drive, BUS-kiértékesítő, lemez-kártya, floppy-tápegység, magnó, 85 gombos IBM klavirútra (beszerelve), 2 db gyári joy-stick, rengeteg szaktudásom.
AZ ÁRBAN MEGEGYÉZUNK!

Cím: Both Anikó, 2800 Tatahánya, I.,
Gál István ltp. 206. 2/5.

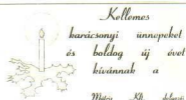
TISZTELT ELŐFIZETŐK! AZ ENTERPRESS NEM SZÚNIK MEG!

Kedvenc újságjótokat 1993. január 1-től az AMEKO Kft. adja ki. Az ENTERPRESS terjedelme 12 oldal lesz és kéthavonta jelenik meg. A lap utcai terjesztése megszűnik, így azt ne keressétek az újságárusoknál! Csak a Kiadó címén lehet megrendelni az ENTERPRESS-t. Egy példány ára 46 Ft lesz. (Sajnos a postai költségek nagyon magasak!) Előfizetés egy évre: 276 Ft, fél évre: 138 Ft. A lap csak akkor tud „indulni”, ha minimum 900 előfizetője van! A jelenlegi előfizetők száma közel 600 (!). Bízunk abban, hogy akik eddig előfizetői voltak az újságnak, azok a jövőben is megrendelik majd az ENTERPRESS-t. Akik eddig a postánál rendeltek meg a lapot ők is csak az új Kiadó címén rendelhetik majd meg. Azt szeretnénk kérni tőletek, hogy népszerűsítsétek a lapot, próbáljátok új előfizetőket „bevásárolni”. Jelenleg az új szerkesztőség minden energiáját a tevékenység emelési fel. A Szerkesztőség továbbra is a régi postacímre érhető el (ENTERPRESS, 1399 Budapest, Pf. 701/334.) A megrendelő lapon rendelhetitek meg az ENTERPRESS 1993. évi számait.

Köszönettel az új szerkesztőség nevében:

Matyas István

A Mikrovilág minden számában két oldalnyi terjedelemben foglal- kozik ENTERPRISE-os témákkal.



LEVELEZÉS

A géppel kapcsolatos témákban levelezze:
Bognár Balázs, 9443 Pécsháza, Bartók Béla u. 11.
Czibere Lajos, 4027 Debrecen, Fiercsi út 1. III/12.
Feczkó Krisztián, 8200 Veszprém, Anyos u. 1/3. Tel.: (80) 29-493
új: Mártóczy Béla, 4400 Nyíregyháza, Új u. 28.
Tóth István, 7305 Mecsekpályás, Szegeft u. 1.

KLUB

Budapesti Enterprise klub VS&M körzetégi klub
Budapest, XI. ker. Fehérvári út 120.

*** ENTERPRESS Kétszemélyes az Enterprise számítógépes felhasználók *** II. évfolyam 5-6. szám *** 1992. szeptember—december *** Kiadó a MATRIX Kft., Székesfehérvár * Feladó kiadó: Juhász István Győző * A kiadó címe: MATRIX Kft., 8000 Székesfehérvár, Zichy liget 10. Telefon (02) 329-880 Telefax (02) 329-880 *** Feladó szerkesztő: Ujvári László * A szerkesztőség tagjai: Hargit Csaba Húszháromi, Devlősné, JÓV, An Gábor, Boszi Gábor, Huskai László, Lóssófi, Mezősoma Gyula * A szerkesztőség csak levelet érthető el a cím: ENTERPRESS, 1399 Budapest, Pf. 701/334. * Technikai szerkesztő: Stoppner László *** Nyomda a Dunántúli Nyomda Kft. * Feladó vezetők: Farkas István *** HIR ISSN 0888-1820 *** Tisztelt a Magyar Posta *** Az ENTERPRESS is megkapja az információs címet az, hogy segítsék, technikai és más feladatok elvégzését. A közlő programokat, kapcsolási rajzokat, képeket modernizációs szövegek felhasználásához, de főleg azokat a kiadó hírellyel engedélyezéssel megkapja. * A szerkesztőség kéziratait nem őzi meg, és nem küld vissza, továbbá kéziratait levélben nem fogadja. *** ENTERPRESS © 1992 MATRIX Kft. ***