

NEXT ↓ CTR G  
 NOT ↓ K  
 ON ↓ !  
 OPEN ↓ "  
 OR ↓ S  
 OUT ↓ space  
 PEEK ↓ e  
 POINT ↓ F  
 POKE ↓ l  
 RANDOM ↓ CTR F  
 READ ↓ CTR K  
 RESET ↓ CTR B  
 RESTORE ↓ CTR P  
 RESUME ↓ ↑  
 RETURN ↓ CTR R  
 RIGHTS ↓ y  
 RND ↓ Ü  
 RUN ↓ CTR N  
 SAVE ↓ -  
 SCREEN ↓ .  
 SET ↓ CTR C  
 SGN ↓ W  
 SIN ↓ b  
 SQR ↓ Á  
 STEP ↓ L  
 STOP ↓ CTR T  
 STR\$ ↓ t  
 STRING\$ ↓ D  
 TAB( ↓ <  
 TAN ↓ c  
 TEST ↓ )  
 THEN ↓ J  
 TO ↓ =  
 TROFF ↓ CTR W  
 TRON ↓ CTR V  
 USING ↓ ?  
 VAL ↓ u  
 VARPTR ↓ É

Hónapokig tartó tanulással lerövidíthető a gépelési idő. A programokon belüli karakterdefiniálás az előbbi szavak képzési módját nem befolyásolja.

A dokumentációellátás igen szegényes. Az ún. felhasználói kézikönyv kritikán aluli: hiányos, megjelenése előtt is elavult, szakmailag hibás. A hardvert és a szoftvert ismerető füzetek adnak ugyan komolyabb betekintést a Primo lelkivilágába, de az alapos ismerethez sok próbálkozás kell. Érvényesül a Murphy-törvény átírata: „Legjobban a káromkodás nyelvét ismerjük”! Jó tanácsom tehát a szalonképességet és az időrablások ellen küzdő gyakorlatlan primósoknak: mielőtt kikezdesz a Primóval, lépj házasságra a HT-vel!

Két kezdő színvonalú programjavaslatom van. Az egyik a „Rajzoló”. A gépkönyv végén közölt ugyanilyen című program az ott leírt módon nem működik. Az 1. lista a négyirányú rajzoló javított változata, tehát nem új. A másik az Élet és tudomány 1985-ös zsebkönyvének csirkevadászati c. feladványa, Primóra átírva (2. lista). Ez sem új. A listák C64-en készültek, „aki” erősen töri a magyart.

Végül két észrevétel. 1. A Primo érti a gépi kódhívás USR-jét. Ezt semmilyen dokumentációban nem találtam, a ROM-ban sem. 2. Az egyik B-32-es gép hibás kazettatöltés utáni programindításkor „S Error” hibajelzést ad, és csak a RESET-re reagál. Nincs rá magyarázatom.

TAKÁCS DEZSŐ

## 2. lista

```

5 CLS:PRINT"CSIRKEVADASZAT"
6 PRINT"A SAS SPACE-SZEL VADASZIK"
7 PRINT"KILEPES "BRK"-VAL"
10 S=1:Z=0:G=1
11 INPUT"A SAS SEBESSEGE(1-10)":V
12 IF V<1 OR V>10 THEN 11
15 CLS:U=(1+V)/2
20 FOR I=0 TO 255:SET(I,0):NEXT
55 C=20+RND(220)
60 FOR I=C-2 TO C+2:SET(I,2):NEXT
90 M=50+RND(100)
100 Y=M:V1=M
105 FOR R=1 TO 255
110 IF U#R>255 THEN CLS:GOTO 20
115 RESET(U*(R-1),M):RESET(U*(R-1),M+1)
120 SET(U#R,M):SET(U#R,M+1)
130 IF INKEY#="" THEN 170
135 FOR Q=0 TO 10:NEXT Q
145 NEXT R
150 FOR J=0 TO 1000:NEXT
155 CLS
160 GOTO 20
170 FOR I=1 TO 300
175 X=U*(R+1)
180 IF X>255 THEN 240
185 RESET(X-U,M):RESET(X-U,M+1)
190 RESET(X-U,V):RESET(X-U,V+1)
195 SET(X,V)
200 Y2=2#V1-Y-G
210 Y=V1:Y1=Y2
215 IF Y<4 AND ABS(C-X)<=3 THEN 260
220 IF Y<1 THEN 240
230 NEXT I
240 PRINT"SEM JOI":S:"LECSAPASBOL":Z:"TALALT"
245 S=S+1
250 GOTO 150
260 Z=Z+1
265 PRINT"JOI":S:"LECSAPASBOL":Z:"TALALT"
266 S=S+1
270 RESET(C-1,2):RESET(C+1,2)
280 GOTO 150

```

## PRIMO

# Gépi kódban

### Értékadás

A G1 jelű BASIC programban (1. lista) lévő gépi kódú rész a HL jelű regiszterben elhelyezi a négy értéket: LD (HL), 04, és visszatér a BASIC programba: RET.

Az 1. sorban a program neve van.

A 100-as sorban a programban lévő skalár változóknak adunk értéket, mert ha a gépi kódú program helyének meghatározása után és működése előtt „új” skalár változót használunk, akkor a program hibásan működik.

A 110-es sorban K% (index) részére helyet foglalunk. E helyre fogjuk elhelyezni a gépi kódú programot. A lefoglalt hely természetesen lehet nagyobb a szükségesnél, de kisebb nem.

A 120-as sorban megkeressük, hogy a gépi kódú program részére lefoglalt hely hol van.

A 129-es sor csak magyarázó rész. Elhagyható.

A 130-as sorban betöltjük a gépi kódú programot.

A 140-es sorban meghívjuk a gépi kódú programot. A gép a HL jelű regiszter címét a C változóban adja meg.

A 150-es sorban kiíratjuk ezt a címet és az ott tárolt értéket.

A program működését kipróbálhatjuk 0 és 255 között lévő egész számokkal. A programot kipróbálás előtt rögzítsük szalagra, mert hiba esetén törölheti magát.

A tanulás kedvéért rontsuk el a programot a következő módon:

— A HL regiszterbe tört számot töltünk.

— A HL regiszterbe negatív számot töltünk.

— A HL regiszterbe 255-nél nagyobb számot töltünk.

— A 120-as és a 140-es számú sorok között új skalár változókat adunk értéket. Például így:

```
135 FOR I=1 TO 100:NEXT I
```

### Adatátadás

A G2 jelű BASIC programban (2. lista) lévő gépi kódú rész adatot kap a DE regiszteren keresztül, áthelyezi az A regiszterbe: LDA, (DE), majd áthelyezi a HL regiszterbe: LD(HL), A és visszatér a BASIC programba: RET. A gép sajátossága, hogy az adatot a DE regiszter fogadja, és a HL regiszter adja vissza.

A 110-es sorban helyet foglalunk az adat számára is.

A 122-es sorban megkeressük, hogy az adat részére lefoglalt hely hol van.

A 132-es sorban a helyére rakjuk az adatot.

A 140-es sorban úgy hívjuk meg a gépi kódú részt, hogy egyben átadjuk az adatot is.

Az 1., 100., 105., 120., 129., 130. és a 150-es sorok a G1 jelű program után nem igényelnek magyarázatot.

Ezt a programot is próbáljuk meg elrontani a G1 programnál leírt módszerekkel!

SOMOGYI GYÖRGY

### 1. lista

```

1 REM G1
100 C=0 : CC=0
110 DIM K%(20)
120 CC=VARPTR(K%(0))
129 REM : LD(HL),04 : RET
130 POKECC,54,4,201
140 C=CALL(CC)
150 PRINT C,PEEK(C)

```

A PROGRAM	A KÓD
LD(HL),04	54,4
RET	201

### 2. lista

```

1 REM G2
100 C=0 : CC=0 : CD=0
105 INPUT Y
110 DIM K%(20),L%(20)
120 CC=VARPTR(K%(0))
122 CD=VARPTR(L%(0))
129 REM : LDA,(DE) : LD(HL),A : RET
130 POKECC,26,119,201
132 POKECD,Y
140 C=CALL(CC,CD)
150 PRINT C,PEEK(C)

```

A PROGRAM	A KÓD
LDA,(DE)	26
LD(HL),A	119
RET	201

## PRIMO

# Gépi kódban

### Értékadás

A G1 jelű BASIC programban (1. lista) lévő gépi kódú rész a HL jelű regiszterben elhelyezi a négy értéket: LD (HL), 04, és visszatér a BASIC programba: RET.

Az 1. sorban a program neve van.

A 100-as sorban a programban lévő skalár változóknak adunk értéket, mert ha a gépi kódú program helyének meghatározása után és működése előtt „új” skalár változót használunk, akkor a program hibásan működik.

A 110-es sorban K% (index) részére helyet foglalunk. E helyre fogjuk elhelyezni a gépi kódú programot. A lefoglalt hely természetesen lehet nagyobb a szükségesnél, de kisebb nem.

A 120-as sorban megkeressük, hogy a gépi kódú program részére lefoglalt hely hol van.

A 129-es sor csak magyarázó rész. Elhagyható.

A 130-as sorban betöltjük a gépi kódú programot.

A 140-es sorban meghívjuk a gépi kódú programot. A gép a HL jelű regiszter címét a C változóban adja meg.

A 150-es sorban kiíratjuk ezt a címet és az ott tárolt értéket.

A program működését kipróbálhatjuk 0 és 255 között lévő egész számokkal. A programot kipróbálás előtt rögzítsük szalagra, mert hiba esetén törölheti magát.

A tanulás kedvéért rontsuk el a programot a következő módokon:

- A HL regiszterbe tört számot töltünk.
- A HL regiszterbe negatív számot töltünk.
- A HL regiszterbe 255-nél nagyobb számot töltünk.
- A 120-as és a 140-es számú sorok között új skalár változónak adunk értéket. Például így:  
135 FOR I=1 TO 100:NEXT I

### Adatátadás

A G2 jelű BASIC programban (2. lista) lévő gépi kódú rész adatot kap a DE regiszteren keresztül, áthelyezi az A regiszterbe: LDA, (DE), majd áthelyezi a HL regiszterbe: LD(HL), A és visszatér a BASIC programba: RET. A gép sajátossága, hogy az adatot a DE regiszter fogadja, és a HL regiszter adja vissza.

A 110-es sorban helyet foglalunk az adat számára is.

A 122-es sorban megkeressük, hogy az adat részére lefoglalt hely hol van.

A 132-es sorban a helyére rakjuk az adatot.

A 140-es sorban úgy hívjuk meg a gépi kódú részt, hogy egyben átadjuk az adatot is.

Az 1., 100., 105., 120., 129., 130. és a 150-es sorok a G1 jelű program után nem igényelnek magyarázatot.

Ezt a programot is próbáljuk meg elrontani a G1 programnál leírt módszerekkel!

SOMOGYI GYÖRGY

1. lista

```

1 REM G1
100 C=0 : CC=0
110 DIM K%(20)
120 CC=VARPTR(K%(0))
129 REM : LD(HL),04 : RET
130 POKECC,54,4,201
140 C=CALL(CC)
150 PRINT C,PEEK(C)

.....
A PROGRAM      .   A KÓD
.....
LD(HL),04      .   54,4
RET            .   201
.....
    
```

2. lista

```

1 REM G2
100 C=0 : CC=0 : CD=0
105 INPUT Y
110 DIM K%(20),L%(20)
120 CC=VARPTR(K%(0))
122 CD=VARPTR(L%(0))
129 REM : LDA,(DE) : LD(HL),A : RET
130 POKECC,26,119,201
132 POKECD,Y
140 C=CALL(CC,CD)
150 PRINT C,PEEK(C)

.....
A PROGRAM      .   A KÓD
.....
LDA,(DE)       .   26
LD(HL),A       .   119
RET            .   201
.....
    
```

PRIMO

## Másolóprogram

Az alábbi, saját készítésű programmal könnyen lehet BASIC és gépi kódú programokat másolni.

Betöltés után a program kiírja parancsait, amelyeket a kezdőbetűkkel választhatunk ki. A másolandó program betöltése után kiírja az indítási címet, amennyiben automatikusan induló gépi kódú program volt az. Bármelyik gomb érintésével visszatérhetünk a menühöz.

A másolóprogramot a BASIC programmal lehet felírni magnóra. Ha ez megvan, LOAD paranccsal tölthetjük be a másolóprogramot, amely automatikusan indul.

A program az eredeti LOAD és SAVE utasításokat átírja, ezért ha már nincs szükség a programra, a tápfeszültség ki-be kapcsolásával hidegindítást kell végezni.

JUHÁSZ ISTVÁN

```

0 ***** Program: Juhász István *****
1 ***** Print :HORZOL 86.10. *****
2 ***** Nev :COPY betolto *****
3 *****
10 CLS
20 PRINT#6,10,"Egy kis türelmet !"
30 FORI=22996TO23517:READA:POKEI,A:NEXT
35 CLS
40 PRINT#2,5,"A magnót allisd felvetelre!"
42 PRINT#4,5,"Ha kész indítsd a magnót."
44 PRINT" majd nyomj meg egy billentyűt !"
46 IFINKEY#="" THEN46ELSEPRINT#9,17,CHR$(2)
"OK"CHR$(18):PRINTCALL(23414)
49 DATA0,0,0,0
50 DATA33,15,69,205,117,43,205,37,0,254,76
51 DATA40,26,254,108,40,22,254,83,202,143
52 DATA68,254,115,209,143,89,254,66,202,20
53 DATA26,254,98,202,25,26,24,223,33,124
54 DATA69,205,117,43,33,92,64,54,131,33,1
55 DATA0,205,143,0,205,117,60,33
60 DATA140,69,119,87,205,158,0,35,119,205
61 DATA158,0,35,119,205,158,0,35,119,205
62 DATA158,0,35,119,71,205,158,0,35,119,16
63 DATA249,205,158,0,35,119,205,117,60,35
64 DATA254,177,40,6,254,185,40,2,24,207
65 DATA119,205,158,0,35,119,205,158
70 DATA0,35,119,205,158,0,35,119,205,158,0
71 DATA35,119,43,43,43,43,126,254,185,194
72 DATA238,67,35,35,94,35,86,235,205,175
73 DATA15,205,102,56,254,0,40,249,195,238
74 DATA67,33,92,64,54,131,33,98,64,34,13
75 DATA69,33,12,69,54,16,205,146,0
80 DATA33,140,69,126,254,177,40,47,254,185
81 DATA40,57,205,221,58,126,205,244,58,35
82 DATA126,205,244,58,35,126,205,244,58,35
83 DATA126,205,244,58,35,70,126,205,244,58
84 DATA35,126,205,244,58,35,16,249,126,205
85 DATA244,58,35,24,204,50,92
90 DATA64,35,126,50,94,64,205,155,0,195
91 DATA238,67,245,205,221,58,241,50,92,64
92 DATA205,244,58,35,126,50,94,64,205,244
93 DATA58,35,126,205,244,58,35,126,205,244
94 DATA58,35,126,205,244,58,195,238,67,0,0
95 DATA0,12,2,32,32,32,32,42,42,42
100 DATA42,42,42,42,42,42,42,42,42,13
101 DATA32,32,32,42,32,32,66,89,46,32
102 DATA74,46,73,46,32,42,13,32,32,42,42
103 DATA42,32,32,67,32,79,32,80,32,89,32
104 DATA32,42,42,42,18,13,13,9,9,32,4
105 DATA76,20,32,79,32,65,32,68,13,13,9,9
106 DATA32
110 DATA4,83,20,32,65,32,86,32,69,13,13,9
111 DATA9,32,4,66,20,32,65,32,83,32,73,32
112 DATA67,13,13,9,9,32,0,73,78,68,73,84
113 DATA93,83,73,32,67,73,77,32,58,32,0
120 DATA33,212,89,17,234,67,1,166,1,237
121 DATA176,33,92,64,54,131,33,205,91,34
122 DATA203,91,33,202,91,54,16,205,146,0
123 DATA33,92,64,54,249,17,238,67,33,146
124 DATA69,1,0,0,205,149,0,22,0,205,221,58
125 DATA62,185,205,244,58,58,94,64,60,205
130 DATA244,58,62,238,205,244,58,62,67,205
131 DATA244,58,62,52,205,244,58,33,25,26
132 DATA229,201,0,0,0,67,79,80,89,32,32,32
133 DATA32,32,32,32,32,32,32,32,0
    
```

## PRIMO — primátus

Keveset írnak a Magazinban a Primóról! (Az októberi szuperakció után remélhetőleg többet...) Engedjenek meg néhány elfogult megjegyzést a gépről.

A SUPER BITLET PRIMO-kínjában olvastam: "... a Primo rövidítve is elfogad BASIC-szavakat, ha megfelelően rövidítjük". Ezt a feltételt paraszti fordításomban kódolásként értelmeztem. A szoftverfüzet tartalmazza a kulcsszó-készlet kódjait, amelyeket a 128-225, ún. definiálható karakterek grafikus jelével hívhatunk elő. Mint tudjuk, a l a vele egyidejűleg nyomott billentyű kódját 128-cal növeli, a CTRL pedig 64-gyel csökkenti. Alkalmazásánál a háromkezű emberek előnyben vannak, ugyanis többször kell egyidejűleg 4 billentyűt is nyomni. Sőt a '84.2 verzió egyenesen zsonglőrökösre készítet, ugyanis a kód módosító billentyűk a klaviatúra két átellenes pontján találhatók. A megjelenített szavakból az is kiderül, hogy ez a kis gép megfelel a szélesebb perifériák követelményeinek, a szoftveres továbbfejlesztésnek. A ROM például tartalmazza a DEF, FN, MERGE, CMD, CVI, CVS, CVD szavakat.

Néhány gyakran használt parancs, utasítás, függvény képzése a következő:

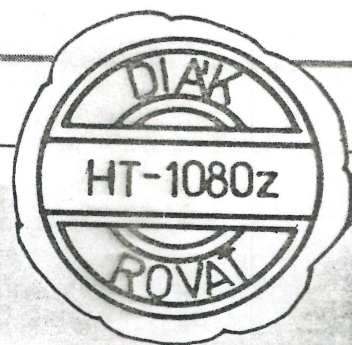
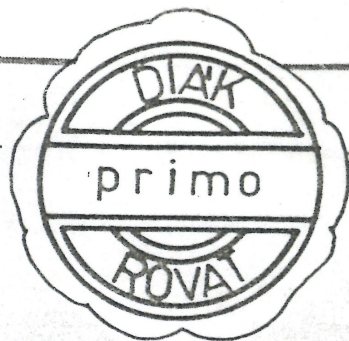
ABS | Y  
AND | R  
ASC | v

ATN | d  
AUTO | 7  
BEEP | CTR Ö  
CALL | A  
CDBL | q  
CHR\$ | w  
CINT | o  
CLEAR | 8  
CLOSE | &  
CLS | CTR D  
CONT | 3  
COS | a  
CREATE | +  
CSGN | p  
DATA | ←  
DEFINT | →  
DEFSGN | CTR Z  
DEFSTR | CTR X  
DELETE | 6  
DIM | CTR J  
EDIT | CTR Á  
ELSE | CTR U  
END | CTR É  
ERL | B  
ERR | C  
ERROR | í  
EXP | é  
FIX | r  
FOR | break  
FRE | Z  
GOSUB | CTR Q  
GOTO | return  
IF | CTR O  
INKEY\$ | I  
INP | ó  
INPUT | CTR I  
INT | X  
LEN | s  
LEFT\$ | x  
LIST | 4  
LOAD | ' |  
LOG | ú  
MID\$ | z  
NEW | ;

### 1. lista

```

10 CLS:PRINTTAB(5) "R A J Z O L O"
20 REM
30 REM
40 REM
50 REM
60 REM
70 REM
80 REM
90 REM SZABÁLYOK SORAI
100 FOR I=0 TO 1000:NEXTI
110 PRINT:PRINT"BARMELY BILLENTYU INDIT"
120 IF INKEY#="" THEN 120
130 CLS:PRINT"KEZDO KOORDINATAK"
140 INPUT X,Y
150 IF X<0 OR X>255 OR Y<0 OR Y>191 THEN PRINT"HIRBA!"GOTO 130
160 CLS:SET(X,Y):S=1
165 AS=INKEY$
170 IF AS="J" AND X<255 THEN X=X+1
180 IF AS="B" AND X>0 THEN X=X-1
190 IF AS="L" AND Y<0 THEN Y=Y-1
200 IF AS="F" AND Y<191 THEN Y=Y+1
210 IF AS="T" THEN 130
220 IF AS="V" AND S=1 THEN S=2
230 IF AS="V" AND S=2 THEN S=1
240 IF AS="S" THEN STOP
250 IF S=1 THEN SET(X,Y)
260 IF S=2 THEN RESIT(X,Y)
270 GOTO 165
    
```



# Tv-foci

A program abban különbözik a korábbi változatoktól, hogy mivel a labda sebessége 1. és 5 között szabályozható, többféle kategóriában teszi lehetővé a játékot. A bal kapust az A és Y billentyűkkel, a jobb kapust az ó és ú billentyűkkel lehet irányítani.

```
1 CLS
5 K=3
10 PRINT$
  1,5,CHR$(3)CHR$(2);"***PRIMO-
  foci***"; CHR$(18)
20 PRINT$
  3,0,STRING$(41,45):PRINT$ 6,2,
  "BAL KAPUS":PRINT$6,28,"JOBB
  KAPUS"
30 PRINT$ 7,3,"FEL:A(a)":PRINT$8,3,
  "LE:Y(y)"
40 PRINT$ 7,29,"FEL:ó":PRINT$ 8,29,
  "LE:ú"
50 PRINT$ 14,8,"NYOMJ MEG EGY
  GOMBOT!!!"
60 IF INKEY$="" THEN 60
65 GOTO 500
70 CLS:PRINT CHR$(1):FOR I= 3 TO
  252: SET(I,1):SET(I,191):NEXT I
80 FOR I=1 TO 55:SET(3,I):SET(252,I):
  NEXT I
90 FOR I=143 TO 191:SET(3,I):SET
  (252,I):NEXT I
100 U=0
110 S=4:W=10
120 N=1
130 PRINT CHR$(6)
140 H=2
150 RANDOM
160 U=U+H
170 C=C+H
180 PRINT$ S,40,CHR$(138)
190 PRINT$ W,0,CHR$(138)
200 PRINT$ C,I,U," "
210 PRINT$ C,U,"o"
220 U1=U
230 C1=C
235 FOR I=0 TO M:NEXT I
240 IF (C=W AND U=0) OR (C=S
  AND U=40) THEN H=-H:GOTO
  160
250 IF U=>40 OR U<=0 THEN
  H=-H
```

```
260 IF C=>14 OR C<=0 THEN
  N=-N
270 F$=INKEY$
275 IF F$="A" AND NOT(W=4) THEN
  PRINT$ W,0," ";:W=W-1
280 IF F$="a" AND NOT(W=4) THEN
  PRINT$ W,0," ";:W=W-1
290 IF F$="Y" AND NOT(W=10)
  THEN PRINT$ W,0," ";:W=W+1
295 IF F$="y" AND NOT(W=10)
  THEN PRINT$ W,0," ";:W=W+1
300 IF F$="ó" AND NOT(S=4) THEN
  PRINT$ S,40," ";:S=S-1
310 IF F$="ú" AND NOT(S=10) THEN
  PRINT$ S,40," ";:S=S+1
320 IF C>=4 AND C<=10 AND (U=0
  OR U=40) THEN PRINT$ C,U,
  " ";:GOTO 340
330 GOTO 160
340 IF U>20 THEN E=E+1 ELSE
  R=R+1
350 PRINT$ 0,0,E," ";:R;CHR$(18)
360 IF E=10 OR R=10 THEN 410
370 IF INKEY$="" THEN 370 ELSE
  PRINT$ 1,13,"15 üres";
380 C=FIX(RND(0)*13+1):IF
  RND(0)>0.25 THEN H=2*SGN(H)
390 GOTO 160
400 CLS:PRINT$ 8,5,"REMÉLEM JÓL
  SZÓRAKOZTAK": GOTO 470
410 FOR I=0 TO 1000:NEXT I
420 CLS:PRINT$ "JÁTSZANAK MÉG?(I/
  N)"
430 F$=INKEY$:IF F$="" THEN 430
440 IF F$="N" THEN 400
450 IF F$="I" THEN RUN 70
470 GOTO 470
490 PRINT CHR$(12):PRINT$ 8,7,CHR
  $(2) "!!! 1...5!!!"CHR$(18):PRINT:
  PRINT:PRINT
500 INPUT$ "MILYEN KATEGÓRIÁ-
  BAN JÁTSZANAK? (1...5)";M
510 IF M<1 OR M>5 THEN 490
520 IF M=2 THEN M=10
530 IF M=3 THEN M=20
540 IF M=4 THEN M=40
550 IF M=5 THEN M=60
560 GOTO 70
```

A program kérésre kiírja az információkat; ezután a változók értékadása, majd a pálya kirajzolása következik. Gombnyomásra elindul a játék. Ketten játszhatják; az egyik játékos ↑ és ↓ nyilakkal, a másik az O és L betűkkel tudja mozgatni „embereit”. Mindkét játékosnak két-két „embere”, azaz ütője van: egy a kapujában, egy pedig az ellenfél térfelének közepén. A két ütőt egyszerre lehet mozgatni. Ha a labda falnak vagy ütőnek ütközik, akkor rugalmasan visszapattan, ha bekerül az egyik kapuba, akkor a legalsó sorban kiírt állás megváltozik: a gólt kapó fél pontja eggyel nő. Ha valamelyik játékos 10 gólt kapott, vége a játéknak, és a gép kiírja a győztest. Ha folytatódik a játék, tehát még nincs 10 gól, akkor a labda a gólt kapó játékos kapujától kerül játékba, mintha a kapus kirúgná.

Esetleg magyarázatra szorul a 390-es és az azt követő sorok szerepe. Miután a program kiszámolta az új koordinátákat (a labdát), megvizsgálja, nincs-e ott ütő vagy fal. Ha igen, akkor a régi koordinátákat számolja ki, a fal vagy ütő elé kirajzolja a labdát, majd egy kis idő múlva leoltja, és az elentétes irányban halad tovább a labda (ütközés). Az 560-as sor a gól után a labda irányát ellentétesre állítja, így az újra játékba kerül.

A labda úgy mozog, hogy az X és Y koordinátáihoz hozzáadjuk U és V változó értékét (2 vagy -2). Ha értékük ellentétes lesz, akkor megfordul a labda útja.

## A program magyarázata

- 10—170 Információ.
- 200—299 A változók értékének megadása és a pálya kirajzolása.
- 300—345 A billentyűzet figyelése, a labda és a játékosok kirajzolása.
- 350 A labda új helyének kiszámítása.
- 360—370 Megvizsgálja, nincs-e gól.
- 380—430 Megvizsgálja, hogy ütközik-e falnak vagy ütőnek a labda.
- 500—560 A gólokkal foglalkozó rész.
- 520—540 Kiírja az állást: ha van 10 kapott gól, akkor vége a játéknak.
- 560— A gólt kapó játékos kapuja felől kerül játékba a labda.

HORVÁTH JÁNOS

```

10 CLS
20 PRINT"FE'L KE'PERNYO !":PRINT
30 INPUT"* KERSZ INFORMACIOT * ";A#
40 IF LEFT$(A#,1)<"I",200 ELSE CLS
50 PRINT"* * * * * * * * * * * *"
60 PRINT"*
70 PRINT"* T U - J A T E K *
80 PRINT"*
90 PRINT"* * * * * * * * * * * *"
100 PRINT:PRINT" 1. JATEKOS: FEL ES LE A"
110 PRINT"          NYILAKKAL MOZOGHAT."
120 PRINT" 2. JATEKOS: 0 - FEL"
130 PRINT"          L - LE"
140 PRINT:PRINT"AZ VESZIT, AKINEK ELOBB LESZ"
150 PRINT"TIZ KAPOTT GOLJA."
160 PRINT:PRINT"MEHET A JATEK ?"
170 IF INKEY#="",170
200 CLS:DEFINT A-Z:A=15810
210 B=15837:X=32:Y=23
220 U=3:V=2:Z=14338:H=14400:C=0
230 FORI=0 TO 62:SET(I,2):SET(I,42):NEXTI
240 FORI=2 TO 9:SET(0,I):SET(62,I):NEXTI
250 FORI=35 TO 42:SET(0,I):SET(62,I):NEXTI
260 FORI=2 TO 42 STEP2:SET(31,I):NEXTI
270 PRINT@960,"HA MEHET, NYOMJON LE VALAMIT";
280 IF INKEY#="",280
290 PRINT@960,"AZ ALLAS:          ";
299 REM * FOPROGRAM *
300 SET(X,Y):K=PEEK(H):L=PEEK(Z)
310 IFK=8 AND A>15488,POKEA,32:POKEA+20,32:A=A-64:GOTO330
320 IFK=16 AND A<16180,POKEA,32:POKEA+20,32:A=A+64
330 IFL=128 AND B>15488,POKEB,32:POKEB-20,32:B=B-64:GOTO345
340 IFL=16 AND B<16200,POKEB,32:POKEB-20,32:B=B+64
345 POKEA,149:POKEB,149:POKEA+20,149:POKEB-20,149
350 RESET(X,Y):X=X+U:Y=Y+V
360 IF X=0 AND Y>9 AND Y<35,500
370 IF X=62 AND Y>9 AND Y<35,510
380 IF Y=3 OR Y=41,U=-U
390 IF POINT(X,Y),X=X-U:Y=Y-V ELSE 300
400 SET(X+U/2,Y+V/2):FORI=1 TO 20:NEXTI
420 RESET(X+U/2,Y+V/2):Y=Y+V:U=-U
430 GOTO 300
500 B1=B1+1:GOTO520
510 B2=B2+1
520 PRINT@970,B1;" - ";B2;
530 C=C+(B2=10)+(B1=10)*2
540 IF C,CLS:PRINT@512,-C;"-ES JATEKOS NYERT !":END
560 X=(SGN(-U))*6+X:U=-U:GOTO 300

```

*A változók szerepe*

		B1	Az 1-es csapat kapott góljainak száma.
A	Az 1-es csapat kapusának a helye.	B2	A 2-es csapat kapott góljainak száma.
A+20	Az 1-es csapat játékosának a helye.	X, Y	A labda jelenlegi X és Y koordinátája.
B	A 2-es csapat kapusának helye.	Z, H	Értékük 14 338 és 14 400; a billentyűzet figyelését könnyítik meg.
B-20	A 2-es csapat játékosának helye.	U, V	Minden alkalommal ezek értéké-
C	A győztes csapat (játékos) száma.		

vel növeljük meg X és Y értékét (értékük 2 vagy -2). Ellentettjük használatakor a labda ellentétes irányba mozog.  
Ciklusváltozó.  
A billentyűzet figyelése alatt értékük PEEK(Z) és PEEK(H) (Z és H értékét lásd feljebb!).

BACZÓ TAMÁS