

Bery Game Programming

Language (© [REDACTED], 1988)

TVC 64K

Ez a program a Basic-hez erősen hasonlító programozási nyelvet valósít meg TV-Computeren. Újdonsága, hogy nem INTERPRETER, hanem COMPILER. Ez azt jelenti, hogy a bevitt utasítássorokat azonnal lefordítja gépi kódra és egy igen hatékony gépi kódú programot állít elő. Eltérése a Basic-től, hogy kifejezetten játékprogramok készítését támogatja.

A program írásakor a megszokott FULL-SCREEN editort használhatunk. Eltérés, hogy a 64. sorhoz és a CTRL-J

nem használható!

A programsorokat sorszámmal kell kezdeni, majd kötelezően egy szóközt tenni. A sorszámkok 1-től 1000-ig terjedhetnek. Ez jelenti egyben a memóriát is. Ha mind az ezer sorban van utasítás, akkor van tele a memória.

A sorszám után az utasítás következik, amit az esetleges paraméter-(ek)-től újabb egy szóköz választ el. Egy sorszámhoz csak egy utasítás tartozik.

Vannak direkt parancsok is. Ezeket nem lehet sorszám után írni. A direkt parancsokat és utasítássorokat RETURN-al kell lezárni. A rendszer sosem ad "OK" üzenetet, csak az esetleges hibák miatt hibüzeneteket.

Csak pozitív, egész számok használata megengedett! A számok 0-tól 65535-ig terjedhetnek.

Változónevek az angol abc betűi lehetnek (csak 1 betű!). Ezenkívül a sprite-ok változói: S1, S2, S3, S4, S5, S6, S7, S8. String típusú változó nem megengedett!

Egy utasítást úgy törölhetünk ki, hogy beírjuk a sorszámát és lezárjuk a RETURN billentyűt. Ekkor -NEM ÉRTEM hibüzenetet kapunk, de a sor ki van törölve, a hibüzenetet nem kell figyelembe venni!

DIREKT UTASÍTÁSOK

NEW

Kitörli a programot és alapállapotba hozza a rendszert. Az esetleges betöltött SPRITE-ok és létrehozott ADATok megmaradnak.

CLS

Letörli a képernyőt.

USR

Elindítja a B.G.P.L. programot, minden esetben az első sorból. Paraméter nem adható meg utánna, ezért ha a programunkat máshol akarjuk elkezdeni, akkor az első sorban el kell ugrani a kívánt sorra.

SAVE 'NÉV'

A B.G.P.L. program kimentése. A név elhagyható, de ha írunk nevet azt mindig csak felső vessző (') közé tesszük! A név maximális hossza 16 karakter. Az utasítás minden esetben kimentti a SPRITE-oknak, ADAT-oknak, szövegelemek és újradefiniálható karakterek fentartott helyét. Ezért rövid program esetén is viszonylag hosszú program file-t ment ki!

LOAD 'NÉV'

B.G.P.L. program töltés. A névre vonatkozó szabályok megegyeznek a SAVE-nél ismertekkel. Ha a töltés megsikad, akkor ismételjük meg, ugyanis esetlegesen tönkretéheti a SPRITE-mozgató és képernyő scrollozó utasításokat! Ez csak futtatásnál derül ki!

SPLD 'NÉV'

SPRITE-töltő utasítás. Az előre elkészített SPRITE-ok beolvasására szolgál. Töltés előtt a rendszer megkérdezi a SPRITE-ok paramétereit.

SPRITE 1 : x, y, m **X**: a SPRITE szélessége pixelben,
:
:
SPRITE 8 : x, y, m **Y**: a SPRITE magassága pixelben,
m: felülírási mód:
0 → teljes felülírás,
1 → XOR szerinti felülírás.

Maximális SPRITE méret:

$$X = 40$$

$$Y = 40$$

DEFC

Kód, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10
Karakter definiáló utasítás. Meggyógyítja a Basic SET CHARACTER utasítással. Kód: az újradefiniálendő karakter kódja (128-182!).

a1-a10: a sorok értékei.

Az elhagyott paramétereket 0-nak értelmezi!

DIREKT UTASÍTÁSOK

LIST

A B.G.P.L. program listázására szolgál.

FORMAI:

LIST

Az összes sort listázza.

LIST 10

A 10. sort listázza.

LIST 10-20

A 10.-től a 20. sorig listáz.

A listázás futásából ESC-vel lehet kilépni!
Adatberítelt szolgáló utasítás.

DATA 1

A program 500 db. 2 byteos adat tárolására szolgál.

2 byte: $\Phi-65535$. Ennek megfelelően az utasítás után paraméterként 1-500 adható meg. A rendszer a RETURN leütése után kiírja az ADATSORSZÁMOT, majd kettőspont után a tartalmát és egy kötőjelet tesz:

1 : 1020 -

A kötőjel után beírható az új adat és RETURN-al bevihető. Ha a régi tartalom megfelelő, akkor a SPACE gomb lenyomására kiírja a következő adatsorszámot és tartalmát. Kilépni bármilyen más gombbal lehet!

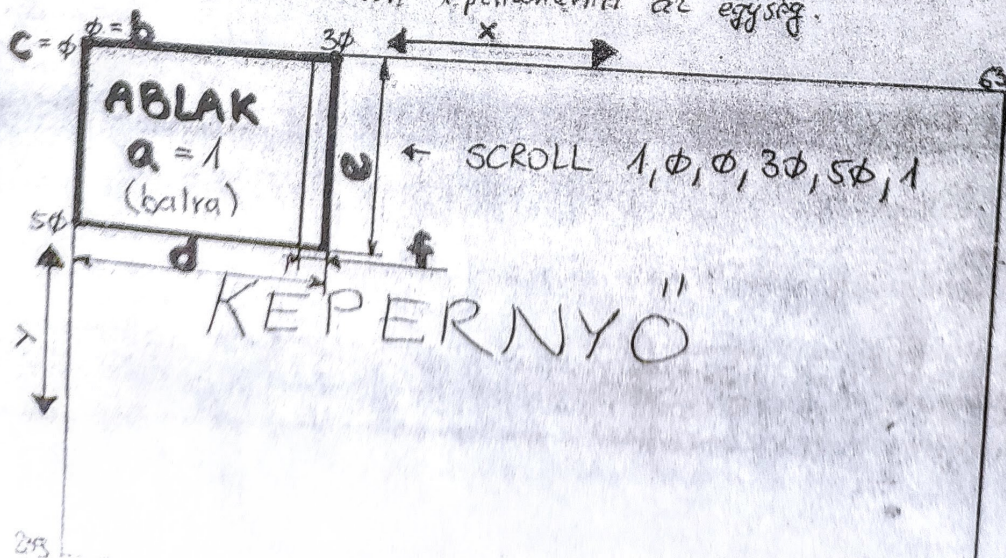
SCROLL a, b, c, d, e, f

A képernyőmozgató utasítások beállítására szolgál.

- a = 1 BALRA SCROLL
- a = 2 JOBBRA SCROLL
- a = 3 LE SCROLL
- a = 4 FEL SCROLL

- b: Az ABLAK x-irányú sarkca ($\Phi-63$).
- c: Az ABLAK y-irányú sarkca ($\Phi-239$).
- d: Az ABLAK x-irányú nagysága ($\Phi-63$).
- e: Az ABLAK y-irányú nagysága ($\Phi-239$).
- f: Egyszeri utasításra elmozdulásra kerülő rész.

Az f BALRA-JOBBRA SCROLL esetén 4 pontonkénti egységben értendő, FEL-LE SCROLL esetén 1 pontonkénti az egység.



A négy irány négy különböző ablakot jelölhet...

UTASÍTÁSOK

PRINT ^{szöveg} Kíró utasítás.

Az utolsó PLOT utasítás jelöli ki a kiírás bal sarkát.
A szöveg hossza 1-32 karakter lehet! Mindig felső vessző ()
közé kell tenni a szöveget. Egy programban maximum
500 karakter lehet kiírni!

PLOT

Vonalhúzó utasítás. Csak 2 paraméter követheti!

PL: 1 PLOT 1000, 2000;

1 PLOT A, B;

Ha pontosvessző (;) zárja ez utasítást, akkor az új pontot
összeköti az előzővel. Ha nincs pontosvessző, akkor nem húz
vonalat! Paraméter lehet szám vagy bármelyik változó (A-Z).
Az első paraméter legyen X, a második Y:

$$0 \leq X \leq 1023, \quad 0 \leq Y \leq 959.$$

PAINT ^{x, y} Festés az adott (x,y) ponttal. x és y csak szám lehet,
lehét változó nem!

CLS Képernyőtörlés.

PALETTE $\phi, \delta\phi, \epsilon\phi, \delta\phi$ Paletta beállító utasítás. Megegyeznie a BASIC
SET PALETTE utasításával. Paraméter csak szám
lehet ($\phi-85$)!

OVER ^x A felülírást állítja PRINT-hez. Egy paraméter követheti:
OVER ϕ - teljes felülírás.
OVER 1 - átlátszó papír.

BORDER ^x A BORDER-t állítja. Értékei: $\phi-85$.

INK ^x Az INK-et állítja. Értékei: $\phi-15$.

PAPER ^x A papírt állítja. Értékei: $\phi-15$.

MODE ^x A felülírási módot állítja:

MODE ϕ - teljes felülírás.

MODE 1 - AND

MODE 2 - OR

MODE 3 - XOR

} szerinti felülírás.

Az INK, PAPER, MODE a PRINT-re vonatkozik csak!

END

viszontérés a fordítóprogramba. Program betöltésre használható
(torés pont)!

- LS - balra
- RS - jobbra
- US - fel
- DS - le

SCROLL

A SCROLL-t kívánt méretű ablakban végzi. Ablak állítása SCROLL utasítással!

SOUND PITCH, DURATION, VOLUME

Hangkeltő utasítás.

- PITCH (hangmagasság): lehet szám: 0-495, vagy **A-Z** változók.
- DURATION (időtartam): lehet szám: 0-255, vagy **A-Z** változók.
- VOLUME (hangerő): lehet szám: 0-255, vagy **A-Z** változók.

PL: SOUND 380, 5, 7 vagy SOUND H, D, 7 vagy ...

GOTO x Ugró utasítás. Paramétere csak szám lehet! $1 \leq x \leq 1000$.

GOSUB x Szubroutineszó utasítás. Paramétere csak szám lehet: $1 \leq x \leq 255$.

RET visszatérés GOSUB után!

REST x A READ utasítások ADATÁRKBÓL való hívásának kezdő ADATCÍME. Csak szám lehet: 1-500.

READ x Adatkiolvasás az adatarból a REST által mutatott kezdő-címtől. Paramétere csak változó lehet: **A-Z**. Az első READ előtt mindig állítsuk be REST-et a kezdő ADATCÍMET!

LDIR a,b,c Másoló utasítás.

- a = honnan
 - b = hová
 - c = hány byteot
- } másoljoni.

FELTÉTELEK:

- $a+c < 65535$
- $b > 6633$

PÉLDA: a képernyő felső részét másoljuk az alsóra.

LDIR 32768, 40448, 768 (VIDEORAM szerint!)

LD a,b,c Feltöltő utasítás.

- a = hová ($a > 6633$)
- b = mennyit ($a+c < 65535$)
- c = mit ($0 \leq c \leq 255$)

SZIN	KÓD
0	0
1	240
2	15
3	255

Példa: töltsük fel a képernyő első 5 karakteres sorát PAPER3 színnel.

LD 32768, 320, 255 (VIDEORAM szerint!)
5*8*4

GET Egy bármilyen gomb lenyomásáig felfüggeszti a program futását!

SP x

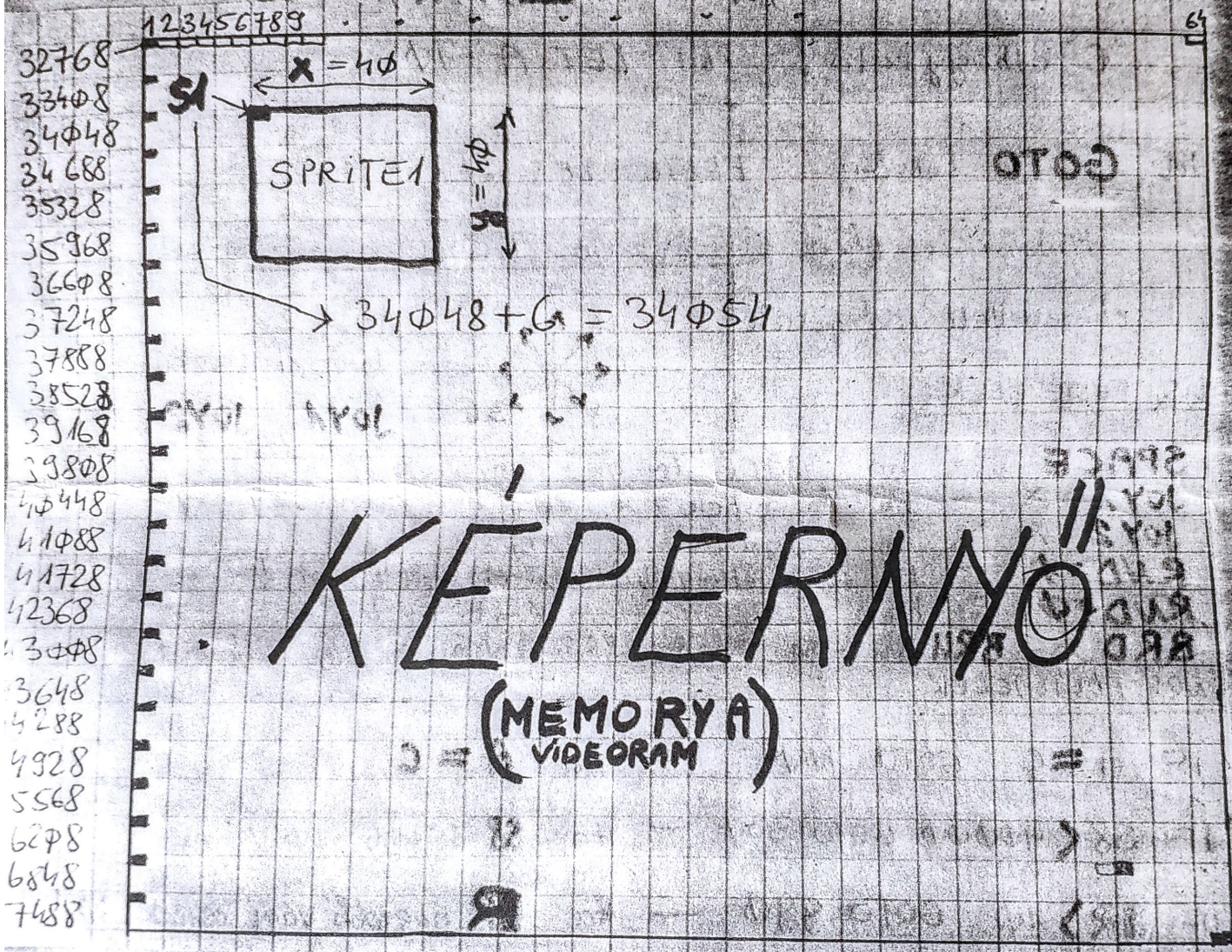
SPRITE kiíró. Csak **SPD** utasítás után van értelme!

$1 \leq x \leq 8$. Minden SPRITE-hoz tartozik egy változó:

SP 1 \rightarrow S1 ... SP 8 \rightarrow S8

A változót 1-el növelve a SPRITE jobbra, 1-el csökkentve balra, 64-el növelve le, 64-el csökkentve fel mozog.

A változót **VIDEORAM** szerint kell beállítani!



PÉLDA: Az 1. SPRITE kiírása.

```

9 GRAPH 1
10 LET S1=34054
11 SP 1
12 GET
13 END

```

GRAPH x

Grafikus módot állítja (GRAPHICS)

- x=1 4 színű üzemmód.
- x=2 16 színű üzemmód.
- x=0 2 színű üzemmód.

LET értékdó, értékállító utasítás!

```

PL:
LET A=1
LET A=A+1
LET A=32φ-B
LET A=A+B
LET X=C
LET J=P-O

```

Tehát összeadni, kivonni és alapértéket beállítani lehet!

```
LET A=B+C
```

A: lehet bármelyik változó.

B: lehet szám és bármelyik változó.

C: lehet szám és bármelyik változó.

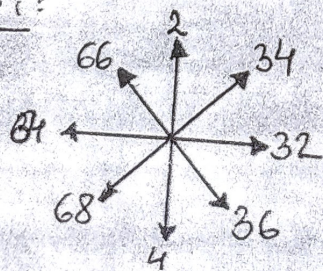
IF feltétel GOTO x

Feltételes elugró utasítás. Mindig GOTO zárja, $1 \leq x \leq 10φφ$.

SPECIALIS FELTÉTELEK:

- IF **SPACE** GOTO y → Ha a SPACE gomb le van nyomva ugorj ...
- IF **JOY1 = X** GOTO y → Ha az első joystick értéke X ugorj ...
- IF **JOY2 = X** GOTO y → Ha a második joystick értéke X ugorj ...
- IF **RND = 1** GOTO y → Ha véletlenváltozó 1 ugorj ...
- IF **RND = φ** GOTO y → Ha véletlenváltozó φ ugorj ...
- ~~IF **BRU = X** GOTO y → Ha X. SPRITE felső sora nem üres ugorj ...~~
- ~~IF **BRD = X** GOTO y → Ha X. SPRITE alsó sora nem üres ugorj ...~~

JOYstick értékei:



TÜZ: 8

INS: JOY1=1

TÜZ + FEL = 8 + 2 = 1φ

BRU, BRD:

(csak 1. φ verzióval lehetnek!)
Az 1.1 verzióval PEEK papíros van beírva!)

a SPRITE-ek alsó, felső ütközését vizsgálja!
Csak akkor működik ha a sprite alsó ill. felső sora üres (háttér=PAPERφ szín). Ha a SPRITE felülírása = φ akkor a háttérütközést nem tudja figyelni, csak XOR felülírással. Úgy kell a beírás sorrendjére. Csak akkor történhet ütközés ha a figyelt SPRITEOT iratjuk ki elsőnek (ez felülírás=φ-nál!), a vele ütközőt pedig utána!

IF feltétel GOTO X

ÁLTALÁNOS FELTÉTELEK:

- IF **A=C** GOTO X Ha **A** egyenlő **C** ugorj X-re!
IF **B<Q** GOTO X Ha **B** kisebb **Q**-nál ugorj X-re!
IF **C>F** GOTO X Ha **C** nagyobb vagy egyenlő **F**-el ugorj X-re!
IF **E/J** GOTO X Ha **E** nem egyenlő **J**-vel ugorj X-re!

Az első paraméter lehet bármelyike változó.

A második paraméter lehet bármelyike változó vagy szám (0-65535).

PEEK

Funkciója megegyezik a BASIC PEEK utasítás funkciójával, azaz adott memóriacím tartalmát (vizsgálja) tölti egy változóba.

PEEK A=S1 Ekkor **S1** változó által mutatott cím tartalmát tölti **A** változóba! Ez 0-255 közötti szám lehet. Ezzel lehet például SPRITE ütközéseket vizsgálni: ha a kiolvasott tartalom nem háttérkód (0), akkor ütközés történt. Arról viszont gondoskodnunk kell, hogy a SPRITE ott egyébként háttérszíne legyen!

Első paramétere lehet **A-Z** változók bármelyike.

Második paraméter lehet **A-Z** és **S1-S8**.