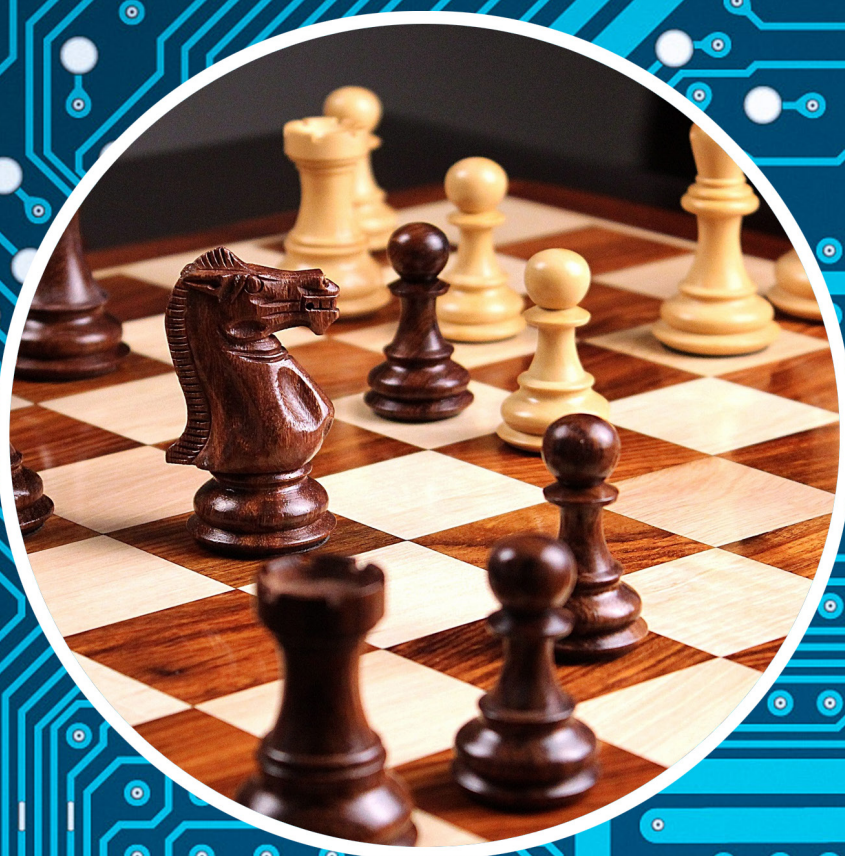


ENTERPRESS

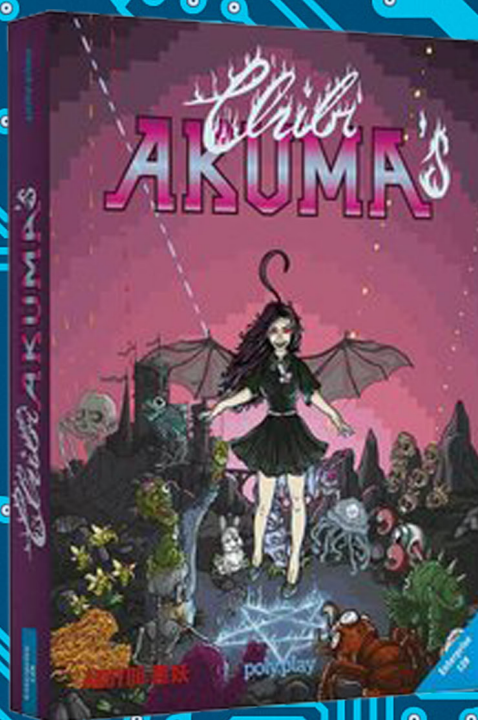
Magazin az ENTERPRISE felhasználóknak

2018/6. november – december



8-bites sakkprogramok

**Chibi Akuma's
30 év után az első
„gyári”, dobozos
játék kiadás EP-re!**



ENTERPRISE

— E X D O S —

ENTERPRISE EXPANDABLE DISK OPERATING SYSTEM



The Enterprise Disk Controller (EXDOS) is one of a range of high quality peripherals for the Enterprise 64 and 128 Home Computers.

The superb specification and flexibility of the controller will enhance your Enterprise and demonstrate that the Enterprise range has *Obsolescence Built Out*.

Features:

- Compatible with any Shugart 410 interface disk drive
- Will drive up to 4 3½" or 5¼" drives and any combination of double or single sided, 40 or 80 track disk
- Fast, random access file retrieval

- Compatibility with MS-DOS file formats allows reading of disks from many popular micro-computers, including the IBM-PC, APRICOT and ATARI 520 ST
- The IS-DOS operating system gives the option of compatibility with C/PM 80 programs
- Full directory, sub directory and 'disk housekeeping' facilities
- And many other advanced features

GIVE YOUR ENTERPRISE THE POWER AND SOPHISTICATION OF A BUSINESS MICRO AT A FRACTION OF THE COST

ENTERPRISE

PERIPHERALS

Enterprise reserves the right to modify details without notice

Enterprise Computers Ltd, 31-37 Hoxton Street,
London N1 6NJ Telephone: 01-739 4282

C/PM is a registered Trade Mark of Digital Research Ltd
MS-DOS is a registered Trade Mark of Microsoft Ltd

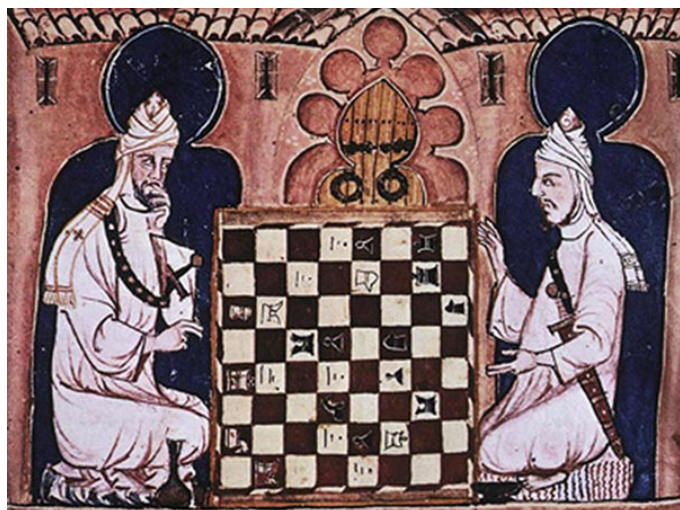
WITH OBSCOLESCENCE BUILT OUT

8-bites sakkprogramok



Írta: Kiss László
(Lacika)

A sakkjáték feltalálásának időpontja a múltba vész, valószínűleg Perzsiából (Irán) vagy Indiából ered. Legrégebbi említése - az ismert „búzaszemes” mese szerint - egy brahmin, név szerint bizonyos Sessa ebn Daher találta fel a sakkot, kb. 1500 évvel ezelőtt. A játék a XIII. században jutott el Európába, ekkor a szabályai még nem voltak egységesek. Az első hivatalos sakkvilágbajnokságot 1886-ban játszották: az első sakkvilágbajnok **Wilhelm Steinitz** volt. A sakk „gépiesítésére” tett első kísérlet 1912-ben a spanyol



feltaláló, **Leonardo Torres y Quevedo** mechanikus szerkeze, amely képes volt bármilyen állásból egy királlyal és egy bástyával mattot adni egy királynak. A gép nem mindig találta meg a legrövidebb lépéssorozatot, de idővel a mattig mindenképpen eljutott. 1944-ben **Neumann János** és **Oskar Morgenstern** publikálja a játékelméletről szóló közös tanulmányát. Eszerint a matematika segítségével minden játék osztályozható, „a sakk a véges-kétszemélyes-nulla-végösszegű teljes információtartalmú játékokhoz sorolandó”. Vagyis lehetséges számítógépes sakkprogramot írni, hiszen a megfelelő algoritmus (elméletileg) képes megtalálni a mindenkori legjobb lépést (a „véges” jelző azt jelenti, hogy véges számú megengedett elrendezés létezik - legalábbis elvileg - lehetséges minden érvényes folytatást előállítani). Ez az algoritmus a Minimax elv, amely a zéró összegű játékok esetében használható. Nemes Tihamér 1949-ben meg is alkotja egy sakkozógép tervét, de az analóg gép sosem épül meg.

Az első - szabányi méretű - elektronikus digitális számítógépet, az ENIAC-ot 1943 és 1945 között „építették”, köve-

tői 1950-es évek elején gyorsan terjedtek. **John McCarthy** pedig megalkotja a „mesterséges intelligencia” kifejezést az első, a témának szentelt konferencián, 1956-ban. Ez nem sci-fi rajongók konferenciája volt (bár az elnevezés eredetileg a tudományos-fantasztikus irodalom terméke), McCarthy a Stanford Egyetem Számítástudományi Karának professzora volt, ő fejlesztette ki a LISP programozási nyelvet. A mesterséges intelligencia a számítógép-tudomány jelentős ágát képviseli, amely intelligens viselkedéssel, gépi tanulással, és a gépek adaptációjával foglalkozik. Szinte triviális, hogy a sakkjáték igen korán a mesterséges intelligenciát kutatók figyelmébe került.

1948-ban **Alan Turing** – aki saját bevallása szerint nem volt erős sakkjátékos, viszont a mesterséges intelligencia kutatása annál inkább foglalkoztatta - kollégájával, **D. G. Champenowne**-nel kezdték fejleszteni az első sakkprogramot, mely 1950-re elkészült, és a készítőitől a Turbochamp nevet kapta. „Apró” gond azért adódott: nem volt megfelelő hardware, ami képes lett volna futtatni a programot, ezért papíron „szimulálták” a program működését (egy lépés „kiszámításához” nagyjából fél óra szükségeltetett). Az első számítógépes sakkprogramra, amely tényleg számítógépen futott, 1956-ig kellett várni: a Los Alamos programot **Paul Stein** és **Mark Wells** fejlesztette MANIAC I számítógéphez. Ugyanebben az évben John McCarthy feltalálta az alfa-béta keresési algoritmust, amely jelentős fejlődést hoz a sakkprogramok terén (is), mivel hatékonyan csökkent a kiértékelendő állások számát. (A játékokban a mesterséges intelligencia működése dióhéjban egy játékfára egyszerűsödik, amelyben az egyes lehetséges lépésekre adott lehetséges lépések szerepelnek körkörösén, ameddig a játék állása a végéhez nem ér. Ebből is látható hogy a fa mélysége óriási méreteket ölthet fel. Éppen ezért ha van olyan lépésünk, amelyet úgysem lépünk meg, mert találtunk már vele egy szinten egy nagyobb hasznosságú lépést, akkor nem vizsgáljuk tovább a sohasem be nem következő lépésünk következményeit.) 1958-ban az NSS az első sakkprogram, amely az alfa-béta keresést használja.

Szögezzük le: az első sakkprogramok képessége inkább csak a szabályos lépkedésben merült ki, 1958-ban győzött le a számítógép embert először: egy IBM 704 számítógépen futó program vert meg egy „átlagos” sakk tudású IBM alkalmazottat. Ekkor még több érdekességet tartogatott két program egymás ellen vívott csatája. 1966-1967-ban két számítógépes program mérte össze tudását: A Moszkvai Elméleti és Kísérleti Fizika Intézet (ITEP) kilenc hónapon keresztül tartó, 4 partiból álló páros mérkőzésen 3-1 arányban, telegráfon kommunikálva legyőzi a Stanford Egyetemen a Kotok-McCarthyt nevű programját. 1974-ben rendezték Stockholmban az első sakkprogram-világbajnokságot, melyet a szovjet Kaissza nevű program nyert meg (4 partiból 4 győzelmet aratva). A moszkvai Vezérléstudományok Intézetének programja egy M-2 nevű számí-

tógépen futott, amely másodpercenként kétezer utasítást volt képes végrehajtani. A versenyen indult a magyar fejlesztésű Papa program is. A szovjet program legfőbb riválisa az elkövetkező években az amerikai Northwestern Egyetem Chess programjának egymást követő verziói. A világbajnokság logisztikai lebonyolításának részletei érdekesek lennének annak fényében, hogy a „bajnok” számítógép 1879 elektroncsövet tartalmazott, 22 négyzetméteres területet foglalt el.

Nyilvánvaló, hogy a '70-es évekbeli programok játékeréje is még mindig igen „szerény volt”. Ennek ellenére a hurráoptimista mesterséges intelligencia kutatók, pl. John McCarthy és Donald Michie azt jósolták, hogy a számítógép 10 éven belül képes lesz megverni a mindenkori sakkvilágbajnokot. **David Levy** (róla még fogunk hallani) nemzetközi sakkmeister viszont 1250 fontos fogadást mert kötni, hogy tíz éven belül nem fogja tudni megverni számítógép. Sőt azt is megjósolta, hogy az ezredforduló előtt nem lesz képes megverni gép a sakkvilágbajnokot. 1977-ben igazolva állítását el is verte a Kaissza aktuális verzióját, majd 1978-ban az amerikai Chess 4.7-et is (a kor két legjobb sakkprogramját). Kevesen gondolták volna tehát akkoriban, hogy 1997-ben az az IBM Deep Blue sakkszámítógépe (a hardvert is speciálisan a sakk számításgényeihez alakították ki) New Yorkban egy hatjátzmás páros mérkőzésen 3,5-2,5 arányban legyőzi **Garri Kaszparovot**, az emberi sakkozás akkori világbajnokát (ehhez átlag 200 millió állás/másodperc kiértékelési sebességre volt szükség). A dolog pikantériája, hogy korábban - azok után, hogy 1985-ben Hamburgban 32 partit játszott a legjobbnak tartott 15 sakkprogram ellen, és valamennyi partit megnyerte - Kaszparov határozott véleménye volt, hogy a gép sosem fogja tudni legyőzni az embert (legalábbis a mindenkori legjobb sakkjátékosokat), de hát azt is tudjuk, hogy „640 Kb-nak mindenre elégnek kell lennie”. A „szkeptikus” David Levy mindössze 3 évet tévedett... No de ne rohanjunk ennyire előre az időben!

A '70-es évek első felében még kevés embernek adatott meg, hogy gép ellen sakkozzon, de 1972-ben az első kereskedelmi forgalomba hozott 8 bites mikroprocesszor az Intel 8008 és követői ezen a téren is új távlatokat nyitottak meg. Számptalan cég kezdett - mai szemmel nézve - megmosolyogtató teljesítményű sakkgépek fejlesztésébe,

forgalmazásába. Két sakkozó: David Levy (vele ugye már találkozunk kicsit feljebb) és Kevin O'Connell megalapította a - sakk könyvek kiadására szakosodott - **Philidor Press Ltd**-et. 1979-ben a cég - haladva az idővel - profilt változtatott: a cég sakkgépek (célszámítógépek) software-jét fejlesztette a SciSys cégnek. Mint ahogy az előző számban már volt róla szó, az első otthoni felhasználásra szánt (megfizethető) számítógépek

1977-ben jelentek meg pár hónap eltéréssel: az Apple II (június), TRS-80 Model I (augusztus), és a Commodore PET (október), kiváló platformot biztosítva a leendő sakkprogramoknak (is).

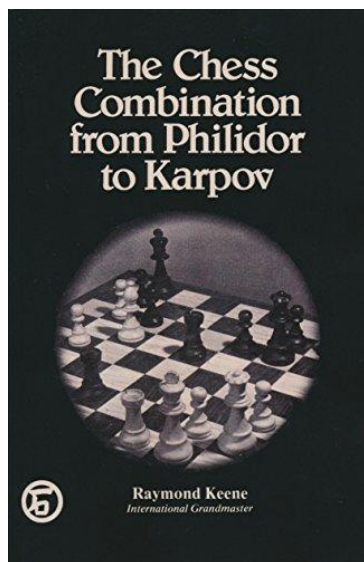
Az első kereskedelmi forgalomba került sakkprogram azonban még a KIM-1 masinériára (ez még nem home computer, „rendes” billentyűzete sincs) 1976-ban megjelent Microchess. A KIM-1 memóriája mindössze 1 Kbyte volt, ebből már sejthető, hogy a program képességei „szerények” voltak. Későbbi változatai azonban megjelentek TRS-80, Apple II, Commodore PET, Atari 400/800 gépekre és Spectrumra is. **Kathe és Dan Spracklen Sargon** című programja lett az



Kathe and Dan Spracklen, ACM 1978

első sikeres sakkjáték sorozat, 1977-ben kezdték fejleszteni. (A program a híres asszír királyról kaphatta a nevét.) Első verziója 1979-ben megnyerte a mikroszámítógépek (8 bites gépek) második sakkversenyét (a rendezvény hivatalos neve PCW Microcomputer Chess Championship volt), 100 százalékos eredménnyel. A program egy Z80 processzorral szerelt Wavemate Jupiter III gépen futott. A program megjegyzésekkel ellátott forráskódja később könyvben is megjelent a Hayden Book Co. kiadó jóvoltából, ezzel sok későbbi sakkprogramozót elindítva pályáján.

A személyi számítógépek térhódítása kihatással volt a sakkgépek forgalmára is: Ahogy zuhant a mikrogépek ára, úgy lett egyre kisebb az árkülönbség a sakkgépekhez képest, amik ráadásul nem is bírtak nagyobb számítási teljesítménnyel: ugyanolyan processzorok dolgoztak a sakkgépekben is. Ezzel egyidőben (a '80-as évek elején) viszont a 8-bites processzorokra fejlesztett sakkprogramok teljesítménye és így története is mind élesebben elvált a nagyszámítógépes társaiktól. Olyan csodák születtek, mint **David Horne 1K ZX Chess** című programja, amely alapkiépítésű ZX81-en is működött, vagyis kevesebb, mint 1Kbyte memóriát igényelt, egészen pontosan 672 byte méretű volt a program. Ezzel 2015-ig a legkisebb méretű működő sakkprogram volt. A Philidor Software rendelései is elapadtak 1981-re, ekkor a cég profilt bővített és nevet váltott: Intelligent Software-re. Az új cég első megrendelője az amerikai - társasjátékokat gyártó - **Milton Bradley** volt. A Milton Bradley egy olyan sakkgépet szeretett volna, ami nem egyszerűen kiírja a gép lépését, hanem egy a saktábla alatt mozgó, elektromágneses (plotterhez hasonló) mechanizmus mozgatja a bábukat. A megrendelő kikötése volt azonban az is,



hogy a fejlesztésnek öt hónap alatt be kell fejeződnie. Bár az IS korábban még sosem foglalkozott elektro-mechanikus megoldásokkal, elvállalta a megbízást (az IS-nek a gép software-jét kellett elkészítenie), mindez történt 1981 júniusában.

1981 szeptemberében, Londonban, a második Mikroszámítógépes Sakk Európa Bajnokságon egy Cyrus nevű sakkprogram nyerte meg (5 játszmából ötöt megnyerve, 100 százalékos eredménnyel), melynek szerzője Richard Lang volt. Lang is a Sargon forráskódjával kezdte az ismerkedést, és jelentősen módosította azt, ennek az eredménye lett a **Cyrus Chess**, amit folyamatosan fejlesztgetve majd minden Z80-as gépre elkészítettek (többek között Enterpreise-ra is). Richard Lang-ot rögtön a bajnokság után szerződtette az IS, akárcsak Martin Bryant-et (a White Knight c. programjával indult ugyanezen a bajnokságon). Ekkor már az IS-nél dolgozott programozóként Mike Johnson, **David Broughton** és **Mark Taylor**. Ezzel a gárdával született meg a Milton Bradley Phantom - Automated Chess Computer. Az IS 1982 januárjára készült el a software-fejlesztésével. Lang 1983-ban kezdte fejleszteni következő - szintén nagy karriert befutó - sakkprogramját: a Psion Chess-t, de ezt nem Z80-ra, hanem a Motorola 68000 processzor-családra.

Az utolsó 8-bites sakkprogram-sorozat az 1986-ban indult Chessmaster. A „sorozat” ebben az esetben már költői túlzás, hisz mindössze a sorozat első darabjának (Chessmaster 2000) van Z80-as verziója, de C64-en is csak a második részig jutott a sorozat (Chessmaster 2100).

Richard Lang a 8-bites éra után is nagyokat alkotott: Chess Genius programja volt az első, ami 1994-ben - igaz csak egyetlen játszmában - meg tudta verni Kaszparovot (ehhez Kaszparov hibája is kellett).

Szerencsére Enterprise-on elég bőnek mondható a futtatható sakkprogramok választéka:

- Cyrus Chess II
- The Chessmaster 2000 (CPC átirat)
- Colossus 4 Chess (Spectrum átirat)
- Sargon 2.1 (CP/M verzió)

Amstrad CPC emulátorral futtatható programok:

- Colossus Chess 4.0
- Masterchess
- Super Chess

Spectrum emulátorral futtatható játékok:

- Cyrus IS Chess
- Masterchess

Felmerülhet a kérdés, melyik sakkprogram milyen erősen játszik? Ezt az első pillantásra könnyen megválaszolhatónak tűnő kérdést nem is olyan egyszerű megválaszolni. Egy sakkprogram játékerejét két módon lehet növelni: Minél több lehetséges állást kell kiértékelni. Több állás természetesen több időt igényel, így állítható az egyes sakkprogramok játékereje.

Az állást kiértékelő algoritmust kell oly módon optimalizálni, hogy minél hamarabb megtalálja azokat az állásokat, melyeket felesleges további mélységben kiértékelni.

Az egyes programokat csak a második szempont szerint érdemes összehasonlítani. Ez azt jelenti, hogy a két program „gondolkodási idejét” kell szinkronba állítani.

1K ZX Chess (ZX81): a lehető legegyszerűbb sakkprogram. A gép mindig világgal játszik, külön indítható verzió van az 4e és d4 kezdőlépésekhez. A sakk szabályait sem tudja tökéletesen: nem ismeri a sáncolást és az En Passant gyakorlatot.

Spectrum Micro Chess (Spectrum): gyakorlatilag véletlenszerűen lépeget ez is, megnyitási adatbázis nincs. Cyruss II (2 fokozat) 19 lépésben mattot ad, miután leszedi a fél táblát.

Spectrum Chess (Spectrum): Úgy tűnik, a Micro Chess-nél csak annyival tud jobban játszani, amennyivel többet gondolkodik, de már van megnyitási adatbázis, 2-3 lépésig... Ez arra elég, hogy valamely felismerhető megnyitásból hatalmas ökörség süljön ki. A két program partija komédiába torkollik, melynek végén a Spectrum Chess nyer.

Spectrum Chess II. (Spectrum): Állítólag erősebben játszik, mint az első rész. Ez a gyakorlatban nem látszik,... Sokszor tanácstalanul lépegetnek egymás ellen, felismerhetően ugyanolyan a „sablon lépéseket” puffogatva.

Chess / Masterchess (Spectrum): Láthatóan más algoritmussal dolgozik, mint a Spectrum Chess, de ez nem jelenti azt, hogy eredményesebb is. Kamikáze stratégiában nyitja az állást, de sáncolni nem nagyon hajlandó (így az egyik bástya jó eséllyel be is ragad). Ez néha még a Spectrum Chess ellen sem működik. Általában a két program eljut egy olyan állásig, amikor mindkettő tanácstalanul lépeget, míg - valamivel gyakrabban - a Spectrum Chess-nek sikerül olyan rosszat lépnie, hogy borul az állás.

A Sargon első verziója 1979 sikeres tudott lenni, de ez az 1981-es 2.1-es verzió nem látszik. Valamivel jobban játszik, mint a Chess / Masterchess, de a középjátékban kiharcolt minőségelőnyt vagy pozícióelőnyt a végjátékban általában nem tudja értékesíteni, és a játszma lépésméltéles döntetlennel végződik. (A Sargon ugyanis már ezt is figyeli, a korábbi programok nem.)

Cyrus IS Chess (Spectrum): Az Intelligent Software programja. David Levy valószínűleg szeret fogadni. A program megjelenésekor épp arra tett fel 5000 fontot egy újságban megjelenő felhívásban, hogy a Cyrus-t egyetlen más (addig megjelent) Spectrum sakkprogram sem tudja megverni. Valószínűleg senki sem tartotta a fogadást, hiszen a megjelenés előtt a program már megnyerte a második európai mikroszámítógépes sakkbajnokságot. A program sem a szolgáltatásokat számítva, sem játékerőben nem egy súlycsoportban játszik az előbbi sakkprogramokkal. Az előbbi programoknál említett „ökörségekért” azonnal büntet. Pozícióelőnyért hajlandó minőséget áldozni. Az ellenfelek a Cyrus tisztjeit vagy gyalogjait kergetik három lépéssel azelőtt, hogy mattot kapnának. A Masterchess 23. lépésig húzza, de „emberi játékosnak” a 12. lépésben illett volna feladni.

Masterchess (level 4) – Cyrus (level 4)

1. d4 d5, 2. c4 Hc6, 3. c4xd5, Vxd5, 4. Hf3, e5, 5. dxe5 Vxd1+, 6. Kxd1, Fg4, 7. Hbd2, O-O-O, 8. Kc2, Fc5, 9. e3, Ff5+, 10. Kc3, Hh6, 11. Hb2, Fb4+, 12. Kc4, Hg4, 13. Hbd4, Fe6+, 14. Hxe6, fxe6, 15. a3, Hxf2, 16. Bg1, Fe7, 17. g4, Bhf8, 18. Fe2,

Bxf3, 19. Fxf3, Hxe5+, 20. Kc3, Hxf3, 21. Bg2, Bd3+, 22. Kc4, a6, 23. Bxf2, He5++

A Cyrus-ban 1-től 8-ig vannak számozva a nehézségi szintek, a Masterchess-ben 0-tól 9-ig, és a Cyrus kevesebbet „gondolkodott”.

Cyrus II. Chess (Enterprise):

Némi algoritmus finomítás vélelmezhető. Különösen gyengébb fokozatokban van játékerő különbség. Zárt állásokat az első rész könnyebben döntetlenre hoz. Nyitottabb állásokat a II rész többnyire megnyeri.



Colossus 4 Chess (Spectrum): 1986-os program, így fiatalabb játékos a mezőnyben. A Cyrus-on kívül az összes eddigi programot simán veri. A Cyrus-t is megizzasztja, valamivel élesebb játékot is bevállal. Ez önmagában dicséretes, de a végjátékhoz közeledve ez egyre öngyilkosabb megoldásokat eredményez. Ha nem sikerül döntő előnyre szert tennie a középjátékban, nagyobb eséllyel a Cyrus nyer.

A Chessmaster-sorozat 1986-ban indult a Chessmaster 2000 programmal. Joggal gondolhatnánk, hogy ez a program játszik az összes felsorolt közül a legerősebben, de a Cyrus II. Chess-hez képest vegyes a kép. Mivel az új kiértékelő algoritmus értelemeszerűen nagyobb számítási igényrel rendelkezik, a gondolkodási időt arányosra állítva, Cyruss II. 1-3-as fokozatban egyértelműen többször nyer. Erősebb (6-8) fokozatokban már fordul a játékerő, de így a Chessmaster irreálisan sokat gondolkodik (olyan sokat, amely versenyjátzmában is elfogadhatatlan lenne, pláne otthon).

Íme egy teljesen vállalható, sakkra emlékeztető „dolog”:

Chessmaster 2000 (level 6) – Cyruss II Chess (level 8)

1. e4 e5, 2. Hf3 Hc6, 3. Fc5 a6, 4. Fa4 Fc5, 5. Fxc6 (kiengedi az előnyét, c3 jobb) dxc6, 6. d3 (passzív lépés, sáncolni jobb) Hf6, 7. Hc3 Fg4, 8. Fe3 Fxe3 (Hf7 jobb, nem ad erős centrumot világosnak), 9. fxe3 Vd6, 10. Vd2 O-O, 11. O-O-O Bfd8 (jobb vezérszárnyon támadni: b5), 12. d4 (bátor...) Ve6 (ez a legjobb!), 13. Ve2 (kötésbe lép, jobb h3) a5 (lassú, jobb b5), 14. Bhf1 a4, 15. dxe5 Bxd1+, 16. Bxd1 Fxf3 (ne szépítsük, ez rossz lépés... Hd7 jobb: tripla izolált gyalogot hagy), 17. gxf3 Vxe5, 18. f4 Vc5 (Ve7 praktikusabb: gyenge e gyalogok érezzék a törődést), 19. Hxa4 Ve7, 20. e5 Bxa4, 21. exf6 Vxf6, 22. a3 Ve7, 23. Vd3 Ba8, 24. e4 Vh4 (felesleges kaland, engedi felszámolni a gyenge gyalogot), 25. Vg3 Ve7 (nem lehet ütni, mert vezércsere után Bd7 következik), 26. e5 Bd8, 27. Vb3 (Bg1 sokkal „hatásosabb”: g6-ra f5 következik) Bxd1+, 28. Kxd1 b5 (súlyos gyengítés! 28 ... Vc5, 29. Vxb7 Vd4+ után örökös sakk tartható), 29. Ke2 Vh4, 30. Vg3 Vd8, 31. Vf3 Vd5 (Hatalmas hiba, teljesen egyenlő állásnál! Pl. Vd7 simán tartja a döntetlent.), 32. Vxd5 cxd5, 33. b3 c5 (Vesztő lépés... Csak Kf8 lehetséges), 34. a4 bxa4, 35. bxa4 Kf8 (késő...), 36. a5 (és a gyalog besétál)

Győzött tehát a Chessmaster, de úgy, hogy irreálisan sokat gondolkodott (értsd: „igazi” gépen bőven több mint 6 óra). A versenyeken használt időlimit az első 40 lépésre 2 óra. A Cyruss viszont 9-es fokozatban is belül van a 2 órán (lehet, hogy direkt így kalkulálták a nehézségi szinteket). A Chessmaster 8-as fokozatban már-már „szemtelenül” pontosan játszik a mezőny többi tagjához viszonyítva, de a válaszdő elfogadhatatlan (még emulátoron, maximális sebességgel is).

A Chessmaster-nek két súlyos hiányossága is van:

Nem ismeri a lépésméltelést döntetlen. (Ez az 1983 előtti programokra volt jellemző...)

És ami már nem is «súlyos», inkább «fatális» és érthetetlen hiba: a program nem engedi, hogy egynél több (azonos színű) vezér, kettőnél több bástya, futó vagy huszár legyen a táblán! Ha egy gyalogot akarunk bevinni és vezért kérnénk, a gép egyszerűen nem hajtja végre a lépést, ha még van vezérünk. Játékállás beállításnál sem állíthatunk be ilyen helyzetet.

Vagyis a program nem pontosan ismeri a sakk szabályait, nem lehet vele minden partit lejátszani!

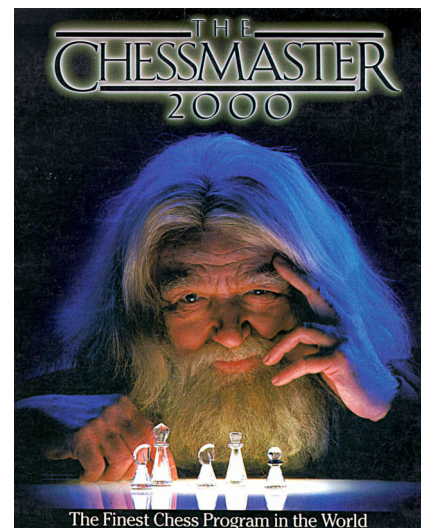
Konklúzióként elmondható, hogy a Cyrus, Chessmaster, Colossus Chess fényévekkel kiemelkedik a mezőnyből. Az összes többi sakkprogram (akármelyik 8 bites platformon is fut) hozzájuk viszonyítva a vicc kategóriába tartozik.

Valamennyi sakkprogramra általánosan igaz: a végjátékban a leggyengébbek. Ez első hallásra furcsának hangzik, hisz ilyenkor van a legkevesebb figura a táblán. De míg a megnyitást majdnem összes program 2-6 lépésig adatbázisból „rakja ki”, végjátékban még így sem lehet teljes elemzést végezni, viszont ilyenkor van a legnagyobb jelentősége a pontos játéknak. Eredményes stratégia tehát bármely programmal szemben, ha az állás egyszerűsítésére játszunk...

A programok másik kényes pontja a zárt állás, amely nagy taktikai érzéket kíván a sakkozótól. Ilyen állásokban, ha elfogynak a kézenfekvő lépések, sablon lépések puffogatása következik, majd előbb-utóbb „beüt” valami sületlenség. Ez még a Cyrus, a Colossus és Chessmaster esetében is megfigyelhető, persze sokkal ritkábban és kevésbé „szemet szűrő” módon.

Azon programoknál melyekből van C64-es verzió, vagy a gondolkodási idő nő (azonos nehézségi szintet beállítva), vagy a pl. Chessmaster esetében (amely már Z80-on is sokat „gondolkodik”) az elemzés mélységét csökkentették (így a CPC verzió nyer).

Összességében szerencsésnek mondhatók az Enterprise tulajdonosok: kezelhetőségben (ide számítva a gondolkodási időt is) egyértelműen a legjobb, játékerőben a második legjobb sakkprogramot futtathatták kedvenc masinájukon. Geco-nak köszönhetően pedig a Chessmaster-ből is van egérrel kezelhető konverzió.



A SOUND utasítás rejtelsei II.

Mit is csinál a STYLE paraméter?



Írta: Bodnár Tamás
(Szipucsu)

A STYLE paraméterrel a hangzás jellegét tudjuk megváltoztatni. A style jelentése stílus, és valóban a megszólalás stílusát, a hangzásformát állíthatjuk be segítségével. Egyetlen paraméter, de többféle effektet (szám szerint hatot) is kezelhetünk vele.

A STYLE által kezelt effekteket érdemes két csoportba osztani:

I. Az egyik csoport effektjei egyetlen csatornán is kifejtik hatásukat, egyetlen SOUND utasítással is bekapcsolhatók. Két ilyen effekt van, az egyik csak a négyszögjelcsatornákon (0, 1 és 2), a másik csak a zajcsatornán (3-as csatorna) használható:

1. háromféle torzítás a három hangcsatornán: SOURCE 0 / 1 / 2 csatornákon STYLE 16 / 32 / 48 értékkel érhető el.

Példák:

SOUND SOURCE 1,STYLE 16 – alacsony torzítás az 1-es csatornán
SOUND SOURCE 2,STYLE 32 – közepes torzítás a 2-es csatornán
SOUND STYLE 48 – magas torzítás a 0-ás csatornán*

*Megjegyzés: A 0-ás csatorna az alapértelmezett, ezért a SOURCE 0 paramétert példáinkban nem adjuk meg külön.

2. ötféle hangzásforma (polinomszámláló bitértékének) megadása a zajcsatornán: SOURCE 3 csatornán alkalmazva STYLE 0 / 4 / 8 / 12 / 16 (alapértelmezés: 0).

Példák:

SOUND SOURCE 3,STYLE 8
SOUND SOURCE 3,STYLE 16

II. A másik csoportba tartozó effektek csak úgy fejtik ki hatásukat, ha két csatornát használunk egyszerre. A STYLE paraméterével ilyenkor egy másik csatornára hivatkozunk (hogy melyik ez a másik csatorna, segédcsatorna,

az effektenként eltér), így ezen a másik csatornán szóló hang magassága befolyásolni fogja az egyik csatornán szóló hang hangzását. A segédcsatorna hangerejét akár 0-ra levehetjük, a hatását akkor is kifejti. Ezek az effektek a következők:

3. a zajcsatorna magasságának változtatása: Korábbi számunkban volt már szó arról, hogy a zajcsatorna magasságát nem lehet megváltoztatni, hiába adunk meg PITCH paramétert. Van azonban egy másik módszer, amivel megváltoztatható a zajcsatorna magassága, mégpedig felveheti egy másik csatorna frekvenciáját: a SOURCE 3 csatornán használva a STYLE 1 / 2 / 3 értéket, attól függően, hogy melyik négyszögjelcsatornának a frekvenciáját akarjuk felhasználni.

Példák:

SOUND SOURCE 3,STYLE 1,SYNC 1
SOUND PITCH 70,SYNC 1,LEFT 0,RIGHT 0 – a 0-ás csatornán szóló hangmagasság adja a zajcsatorna frekvenciáját.

SOUND SOURCE 3,STYLE 2,SYNC 1
SOUND PITCH 100,SOURCE 1,SYNC 1,LEFT 0,RIGHT 0 – az 1-es csatornán szóló hangmagasság adja a zajcsatorna frekvenciáját.

4. gyűrűmoduláció:

a) Gyűrűmoduláció a 0-ás és a 2-es csatorna között: A SOURCE 0 vagy a SOURCE 2 csatornán kiadva a STYLE 128 paramétert, a 0-ás csatorna a 2-es, a 2-es a 0-ás csatorna magasságát használja fel a gyűrűmodulációhoz.

Példa:

SOUND PITCH 25,SYNC 1,STYLE 128
SOUND PITCH 25.2,SYNC 1,SOURCE 2

b) Gyűrűmoduláció az 1-es és a 3-as csatorna között: Az 1-es és 3-as csatornán is kiadhatjuk a STYLE 128 paramétert, így akkor azok egymással hoznak létre gyűrűmodulációt.

Példa:

SOUND SYNC 1,STYLE 128,SOURCE 1, PITCH 49
SOUND SYNC 1,SOURCE 3

5. felüláteresztő szűrő: A STYLE 64 paraméterrel érhető el bármelyik csatornán, és a nála 1-gyel nagyobb számú csatorna frekvenciáját használja fel a felüláteresztő szűrőhöz. Ha a 3-as csatornán (zajcsatorna) adjuk ki, akkor az a 0-ás csatorna frekvenciáját használja fel.

Példák:

SOUND STYLE 64,SYNC 1,PITCH 37
SOUND SOURCE 1,SYNC 1,PITCH 44
– 0-ás és 1-es csatorna között

SOUND SOURCE 1,STYLE 64,SYNC 1,PITCH 37
SOUND SOURCE 2,SYNC 1,PITCH 44
– 1-es és 2-es csatorna között

SOUND SOURCE 2,SYNC 1,STYLE 64,PITCH 49

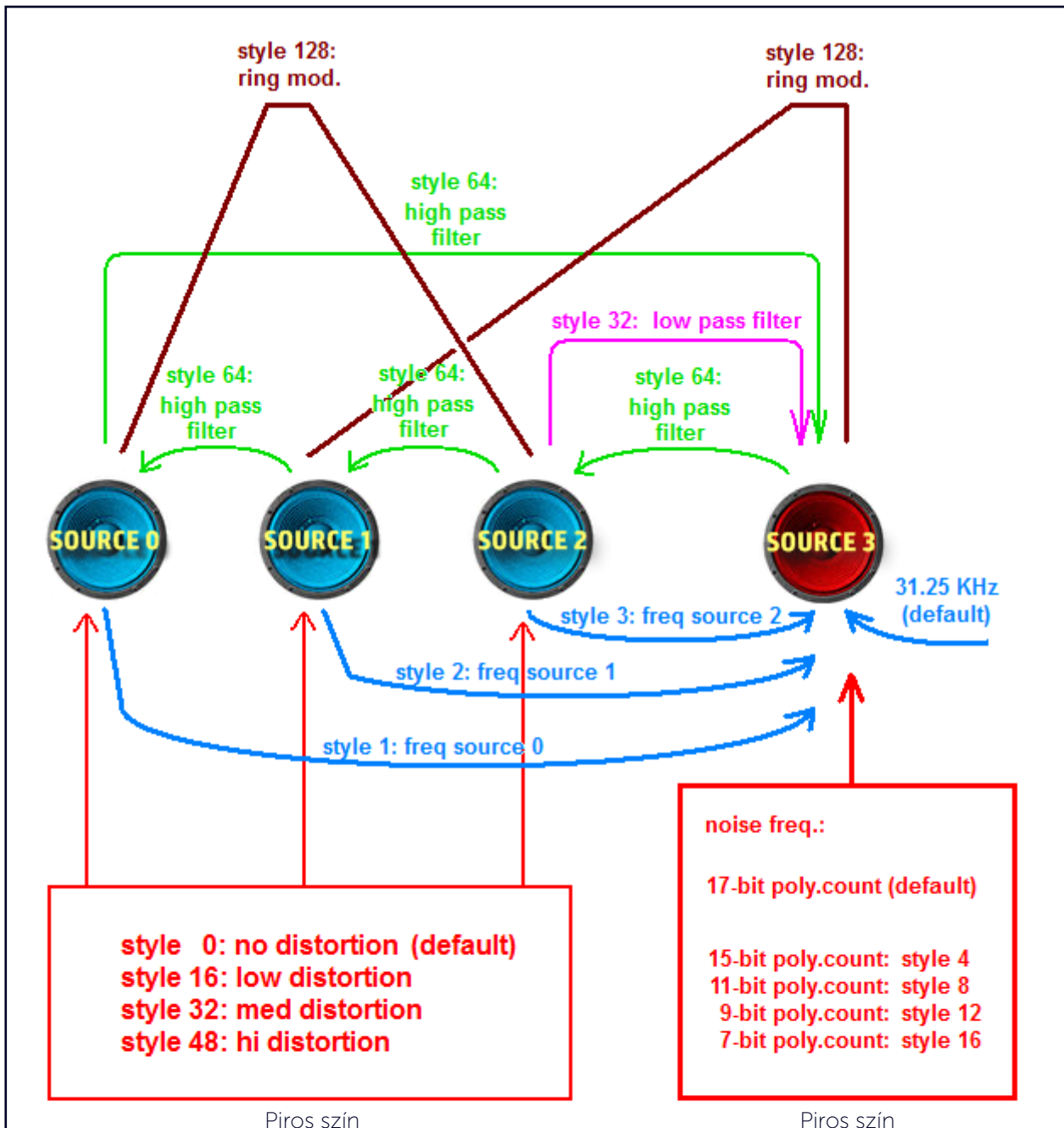
SOUND SOURCE 3,SYNC 1,LEFT 0,RIGHT 0 – a 2-es csatorna és a zajcsatorna között

6. aluláteresztő szűrő: csak a zajcsatornán (SOURCE 3) alkalmazható, a STYLE 32 paraméter megadásával, és mindig a 2-es négyszögjelcsatorna frekvenciáját használja fel.

Példa:

SOUND SOURCE 3,STYLE 32,SYNC 1
SOUND SOURCE 2,PITCH 90,SYNC 1,LEFT 0,RIGHT 0

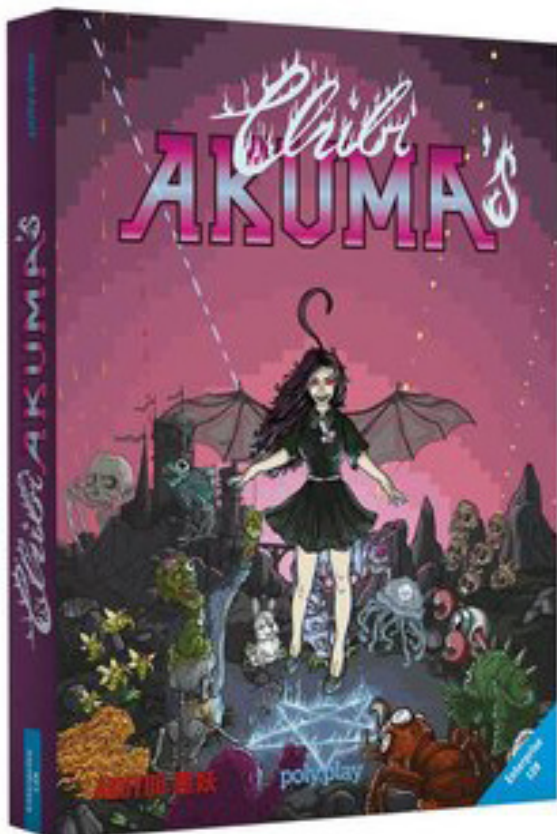
A különböző effektekről bővebben következő számunkban lesz szó.



A SOUND utasítás STYLE paraméterének a működése. Az egyetlen csatornát igénylő effektek pirossal, a kétcsatornás effektek egyéb színekkel vannak feltüntetve.

Chibi Akuma's

30 év után az első „gyári”, dobozos játék kiadás EP-re!



Chibiko utóélete nem túl rózsás. Isteni közbeavatkozással kinyíftatták, az alvilágból is száműzték, ez még csak hagyján. Ami ennél is rosszabb: leírhatatlanul ronda szörnyek árasztották el a birodalmát, ezzel nem kis kalamajkát okozva a kastélyában! Csibiko nem az a fajta vámpír, amelyik csak úgy hagyja az ilyesmit! Ideje kikászálódni a sírból és gonosz varázslatokat bevetve lenullázni a szörnyeket, hogy írmagjuk se maradjon!

Dobozos változat megrendelése:

https://www.polyplay.xyz/Chibi-Akumas-Collectors-Edition-Enterprise-35-Diskette_1

További információk:

<http://chibiakumas.com/>

Nyan Cat

Xmas Nyan Cat Enterprise 128

Povi gondoskodott róla, hogy Enterprise számítógépre is legyen Nyan Cat! Sőt, extra karácsonyi változatot is kaptunk a demóból! Ez látható a képen (majd az elektronikus Enterpressben láthatjátok a színeket :)).

A demó itt látható:

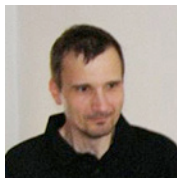
<https://www.youtube.com/watch?v=rLh4zAROfTo>

Szipucsu pedig MIDI-ben készítette el a Nyan Cat zenéjét.



Arrow of Death Part 2

1983 - Digital Fantasia
kaland, szöveges



Írta: Kiss László
(Lacika)

Az [Arrow of Death](#) első részében felkutattuk a Halál Nyilának elkészítéséhez szükséges „alkatrészeket”, a második részében feladatunk a nyílkészítő mester megkeresése (faját és házát felégették, vélhetően raboskodik valahol). Miután elkészíti nekünk a Halál Nyilát, meg kell ölnünk vele Xerdon-t, a gonosz varázslót. Mivel a teljes Mysterios Sorozat ugyanazt a „motort” használja a program kezelése teljes egészében megegyezik a sorozat első darabjának ([The Golden Baton](#)) ismertetőjében leírtakkal.

A program szókészlete:

ALTAR	DYNAMITE	MOUNTAIN	STAIR
ANIMAL	EAT	MUD	STONE
ARCHWAY	EXAMINE	MULE	TAKE
ARNID	FEATHERS	MUSIC	TAPESTRY
ARROW	FEED	OARS	TIE
ARROWHEAD	FILL	OPENING	TO
ATTACK	FLAME	ORGAN	TOBACCO
BOAT	FLINT	PIPE	TRAIL
BOW	FUSE	PLAY	TURN
BRANCH	GORGE	POLISH	UNBOLT
BREAD	GUARD	PRAY	UNIFORM
BREAK	HELMET	PRESS	UNLIGHT
BRIDGE	HOLD	PULL	UNLOCK
BUTTON	HUT	REMOVE	VEIL
CAIRN	JETTY	ROCK	WAIT
CANDLE	JUMP	ROPES	WARRIOR
CAVE	KEY	ROW	WATER
CHEESE	KILL	RUB	WEAR
CLOAK	KITE	SHOOT	WEED
CORRIDOR	LAKE	SHOVEL	WHEEL
CREVICE	LAMP	SHRUB	WINDOW
CUT	LEVER	SKELETON	XERDON
DIG	LIGHT	SLITS	
DOOR	LOOK	SMASH	
DROP	MAKE	SMOKE	

A HELP parancs annyit árul el, hogy keressük meg Arnid-ot, a nyílkészítő mestert, majd a Halál Nyilával öljük meg a gonosz Xerdon-t.

A megoldás:

A játék ott kezdődik, ahol az első rész véget ért: a mocsár szélén állunk (1. kép), és az INVENTORY parancs kiadásával



pedig meggyőződhetünk róla, hogy nálunk van a nyílhegy, mágikus fűzfavessző, sastoll és a kard. Mehetünk keletnek vagy délnek. Északra egy mély szakadék állja utunkat, induljunk inkább délre (E, E). Unalmas síkságon kutyagolunk, de most irányt váltunk (N, N). Egy szűk szurdokba, egy hasadékból víz csörgedezik, ráadásul zsákutcába jutottunk. Rejtjük el itt a nyílhegy alkatrészeit, mert eléggé fel vagyunk pakolva (DROP ARROWHEAD, DROP BRANCH, DROP FEATHERS). Kutyagoljunk még egy kicsit a síkságon (S, S, W, N), a táj sivárságát egy bokor töri meg (EXAMINE SHRUB), ami puha, bolyhos (GET SHRUB). Barangoljunk tovább, ezúttal délnek (S, S). Itt látszólag semmi különös nincs, de ha ásunk (DIG), egy kovakövet találunk (GET FLINT) (ezt mégis hogyan kellene kitalálnunk, még ásónk sincs...?).

A mocsártól északra (N, W, N, N) egy szakadék van - kíváncsiak talán már meg is nézték a játék elején. Semmit nem látunk a szakadékon kívül, de azért nézzünk jobban körül (LOOK DOWN). Egy sziklaszirt van alattunk. Bátor akrobatamutatókövetkezik (JUMP), pont egy híd mellé huppanunk (GO BRIDGE, N). A híd túloldalán egy rakat gázt találunk (GET WEED), micsoda zsákmány... A híd déli oldaláról (S, S), lefelé (D) egy sisak hever a sziklapárkányon, vegyük magunkra (GET HELMET, WEAR HELMET). Egy sziklahasadékot is látunk, menjünk be (GO CREVICE), szerencsére nincs bent sötét, sőt egy olajlámpát találunk (GET LAMP). Újabb „beteg” manőver következik: menjünk fel a hídra (N, U, GO BRIDGE), ahol pár kötélt szabadon „lifeg”. Kapaszkodjunk a kötelekbe (HOLD ROPES), majd a karddal vágjuk el a hidat tartó köteleket (CUT ROPES). Egy sziklabemélyedésre lendültünk a köteleken, mellettünk egy barlang bejárata (ha nem kapaszkodunk meg, mielőtt elvágjuk a köteleket, a mélybe zuhanunk). Ide zuhant pár darab kötélt (GET ROPE), majd menjünk be - persze meggyújtott lámpával (kovakő nélkül nem tudjuk meggyújtani a lámpát) (LIGHT LAMP, GO ARCHWAY). Délre van a kijárat, északra és keletre van járat. Induljunk elő-

szőr északnak (N, N, - fel a lépcsőn - U). Egy apró szobába jutottunk, ahol egy halott harcos hever a földön, vizsgáljuk át (EXAMINE WARRIOR). Megállapíthatjuk, hogy egyenruhát visel, öltünk is magunkra (GET UNIFORM, WEAR UNIFORM). Menjünk le a lépcsőn (D, D), egy hatalmas sziklakamrába jutunk. Egy méretes vaskerek van a falon, előttünk meg egy morcos őr (2. kép). Nem igazán vagyunk egy súlycsoportban vele, ha megpróbáljuk megtámadni, egyetlen dőféssel megöl. Ha megvizsgáljuk a kereket, semmit nem segít a program, de próbálhatjuk megtekerni (TURN WHEEL). Nem vagyunk elég erősek... Újabb hatalmas kiszúrás: Úgy tűnik a gaz, amit találtunk, valami gyógynövény lehet, mert ha megesszük (EAT WEED), erősebbek leszünk (ha megvizsgáljuk a gázt, semmi érdekeset nem látunk rajta). Ha most újra megpróbáljuk (TURN WHEEL), egy morajló hang jelzi, hogy valamit működésbe hoztunk. Ha nincs rajtunk az egyenruha, az őr nem enged minket kereket tekergetni. Menjünk vissza fel, a kis szobába (U, U), vegyük fel a papírsárkányt (GET KITE), majd nyugatra (W) egy hasadék fölé nyúló platóra sétálunk. Vakmerő mutatvánnyal (JUMP) a mély szakadék túlsó szélére ugunk (ha rajtunk van a sisak élve, egyéb esetben beverjük a fejünket és meghalunk...).

Csak délre indulhatunk, de innen visszajuthatunk a mocsárhoz, illetve oda, ahol lepakoltuk „kincseinket” (UNLIGHT LAMP, S, S, E, E, N, N). Pakoljunk le (REMOVE HELMET, DROP HELMET, REMOVE UNIFORM, DROP UNIFORM, DROP KITE), majd fel (GET ARROWHEAD, GET BRANCH, GET FEATHERS). A nagy pakolás közben a figyelmesebbek észrevehették, hogy valami megváltozott: a vaskerekkel az itt kifolyó víz útját zártuk el, és most a hasadék helyén egy nagy sárkupac van. Dagonyázzunk egy kicsit (GO MUD), ha alaposabban „megmártózzunk” a sárkupacban (EXAMINE MUD), egy fogantyút tapintunk ki. A teendő talán egyértelmű: húzzuk meg (PULL LEVER). A sziklafalban ezzel kinyitottunk egy kőajtót. Menjünk is be (LIGHT LAMP, GO OPENING).



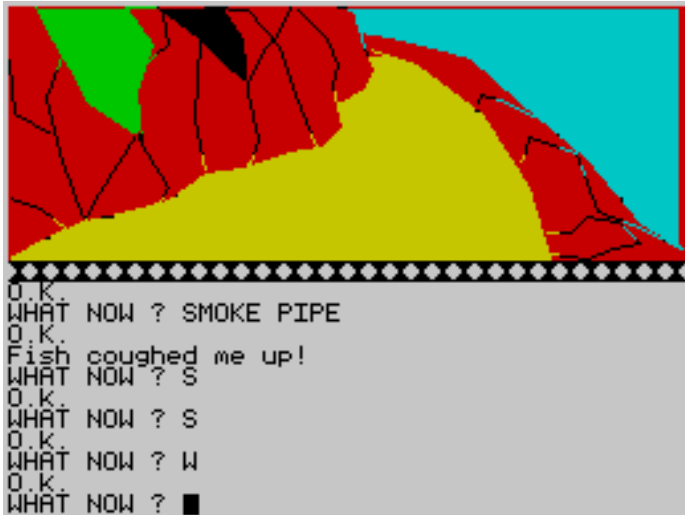
Egy előcsarnokba jutottunk ahonnan, mehetünk keletre, vagy lefelé, nyugatra a kijárat van. Induljunk el a keleti folyosón (E, N). Az örök pihenőszobájában - ha őrral nem is - egy furcsa állattal kerülünk szembe. A szeme sem áll jól (KILL ANIMAL, vagy ATTACK ANIMAL). Megölni ugyan nem tudjuk, csak elkábul, viszont elejt valamit: egy kulcsot! Több cuccot nem bírunk el, a kardod lerakhatjuk (DROP SWORD, GET KEY). Délre indulva egy kisebb labirintusban kóvályog-

hatunk (S, E, S, - egy lépcsőt találtunk - U). A lépcső tetején egy bereteselt ajtó állja utunkat. A „balga” játékos hiába próbálkozik teljesen logikus dolgokkal, OPEN, UNLOCK hatástalan, csak az (UNBOLT DOOR) vezet eredményre... (Az ajtót csak erről az oldalról lehet kinyitni.) Az ajtó mögött - a változatosság kedvéért a folyosók szövevényén ismerős helyre jutunk (GO DOOR, W) a morcos őrhöz, innen



kijuthatunk a labirintus bejáratához, majd benézünk keletre (U, S, S, E, E). A folyosóról itt egy ajtó nyílik, pontosabban nyílha, ha nem lenne zárva. A begyűjtött kulccsal viszont kinyithatjuk (UNLOCK DOOR, GO DOOR). Egy élelmiszerraktárt találtunk, ahol magunkhoz vehetünk némi elemőzsiát, mondjuk a kenyeret (DROP KEY, GET BREAD). Most az előző útvonalon mehetünk vissza, míg egy északi járat végén egy istállóba jutunk (N, W, S, N, N, D, E, GO DOOR, D, N, N). Egyetlen öszvér lakik itt, aki ráadásul elég sovány és éhesnek néz ki, etessük meg (FEED MULE). Az öszvér tele hassal mindjárt vidámabb lesz és követni fog minket (S, W, W - az előcsarnokban vagyunk - , D). Kellemes helyre jutottunk, a várbörtönbe. Egy lerácsozott lyukat látunk. Nem tudjuk kinyitni, nem vagyunk elég erősek. De talán az öszvér! A kötél egyik végét a rácshoz, a másik végét a szamárhoz kötjük, majd meghúzzuk a szamarat (TIE ROPE, TO GRATING, TIE ROPE, TO MULE, PULL MULE). A szamár nekiiramodik, és lerántja rácsot a lyukról, mehetünk lefelé (GO HOLE). A lyuk valójában egy cella, itt raboskodik Arnid a nyílkészítő mester. Ha megvizsgáljuk (EXAMINE ARNID), megtudjuk, hogy eszméletlen. Ez jól félrevezeti a „balga” kalandort, és elrohan elsősegélydobozt, vagy hasonlót keresni... Érdekes módon, ha adunk neki az utolsó adag „gyógygázból” (GIVE WEED, TO ARNID), feléled és hálája jeléül, felajánlja segítségét. Háláját legpraktikusabban a nyílvesző elkészítésével fejezhetné ki (MAKE ARROW) - logikusan az ember elkezdene odaadogatni az „alkatrészeket”, de ezt nem érti a program. Gyorsan végez is a munkával, majd lelécel. A tárgyaink listájában az nyílhegy, fűzfavessző, és a sastoll helyett a mágikus nyílvesző jelenik meg. Azt az apró, de el nem hanyagolható problémát, hogy a nyílveszőhöz nyíl is szükségeltetik, még kezelniük kellene.

Barangoljunk tovább a folyosókon (U, U, E, E, S, S). Egy templomba jutottunk (3. kép). A program felhívja a figyelmünket a falikárpitra, nézzük meg alaposabban (EXAMINE TAPESTRY). Tűzön járó embereket ábrázol és egy ezüst



nyomógombot találunk rajta, amit ha megnyomunk (PRESS BUTTON), egy oltár jelenik meg. Azt is vizsgáljuk meg (LOOK ALTAR). Egy gyertyát találunk rajta, amit meg is gyújtunk a kovakővel (LIGHT CANDLE), megjelenik egy mágikus tűzoszlop. Ha megvizsgáljuk a lángot (EXAMINE FLAME), fantasztikus tudományos felfedezést teszünk: „forrónak tűnik!” A vakmerő kalandoroknak egy kis „meleg” meg se kottyán, és miközben átsétálnak rajta (GO FLAME) porig égnek. A megfontolt kalandor viszont keres egy porral oltó készületet. Nem fog találni... Mit csináljunk egy templomban? Imádkozzunk (PRAY)! A program „segítőkészen”(??) közli „Valami történik...” Ha újra megnézzük a lángot (EXAMINE FLAME), haloványabbnak tűnik. A vakmerő kalandorok most már belegyalogolhatnak a lángoszlopba (GO FLAME). Ezzel a mutatvánnyal egy másik, hegy tetején álló oltárra teleportáltuk magunkat. Sétájunk egy kicsit, lássuk, hova kerültünk (UNLIGHT LAMP, S, S, E). Legyalogoltunk a hegyről, elhaladtunk egy tó mellett, és most egy konyhá mellett állunk. Menjünk is be (GO HUT). A házikóban mindössze egy pipát és némi dohányt találunk. Vegyük fel mindkettőt, és töltsük meg a pipát (GET PIPE, GET TOBACCO, FILL PIPE). Más dolgunk itt nem lévén menjünk és sétálgassunk a tóparton (N, W, S). Találunk is egy csónakot, csónakázzunk egyet (GO BOAT). Látjuk a mólót és az evezőket. Csónakázáshoz az evezők hasznosnak tűnnek (GET OARS). Egy csónakban az ember valószínűleg a PULL OARS, PULLING OARS parancsokkal próbálkozna, de a program csak a (ROW BOAT) parancsot érti meg. A csónak mindenesetre elindul. A csónakázás nem tart sokáig, mert egy óriási hal lenyel minket (LIGHT LAMP). (A hal gyomrát ábrázoló képet nevezzük mondjuk „nonfiguratívnak”, bár ezzel nagymértékben szépítem a tényeket.) Találunk itt egy rongyos köpönyeget - talán a hal valamelyik korábbi áldozatáé. Pinocchio kalandjain felnőve valószínűleg mindenki tüzet akarna gyújtani, de pl. LIGHT LEAVES, BURN CLOAK parancsokat kiadva nem érti a program, mit akarunk. Marad az ifjúság nevelése szempontjából „legártalmasabb” megoldás (LIGHT PIPE, SMOKE PIPE) - füstöt látunk a hal gyomrában, na még egyszer - (SMOKE PIPE). A füstkoncentráció elérte azt a szintet, hogy a hal „kiköhög” a tópartra. Erős a gyanú, hogy csónak hiányában az evezőre már nem lesz szükségünk (DROP OARS). Csak egy irányba mehetünk, ezen legalább nem kell sokat gondolkodni (S, S, W). Egy hegyi ösvény végéhez értünk (4. kép), viszont talá-

lunk egy nagy sziklát és egy barlangot (GO CAVE, N). A barlang végébe értünk, egy kupac kő hever előttünk (mondjuk ez egy barlangban nem annyira rendkívüli...). Ha megvizsgáljuk a köveket, a program semmi érdekeset nem említ, mert a „kupacot” kell megvizsgálni (EXAMINE CAIRN)... Egy csillogó kő kerül elő a kupacból (GET STONE), amit, ha megvizsgáljuk (LOOK STONE), azt mondja a program, hogy „megmutatja a koldust”. Talán sejthető ennek módja (RUB STONE, vagy POLISH STONE), és valóban előttünk terem az ismerős öreg koldus. Adjunk neki valamit, hátha most is meghálálja (GIVE STONE). (Mást nem is enged a program odaadni.) Nem csalódunk a koldusban, egy mágikus nyilat ad hálából, majd eltűnik (GET BOW). Kifele menet vegyük fel az ásót (S, GET SHOVEL), majd ássunk (DIG). Hogy miért pont itt, ki tudja, - szerencsére sok helyen nem is próbálkozhatunk - de találunk egy rúd dinamitot, rövid kanóccal (GET DYNAMITE). Menjünk ki a barlangból (E), ahol a nagy szikla van. Minden épeszű ember elkezd találgatni, hogyan robbanthatná fel a sziklát, esetleg tolná, görgetné, de nem... fel kell venni (GET ROCK)! Másfelé nem tudunk menni, csak a tópartra (E, N, N), időközben idekerült egy izgó-mozgó csontváz. A program szerint egy mozgó csontvázon - aki ráadásul nem enged tovább - nincs semmi különös... Az érthető, hogy egy csontvázat nem lehet megölni, de az ATTACK SKELETON, THROW SKELETON sem működik, csak a (SMASH SKELETON) paranccsal törhetjük össze a csontvázat. Így már mehetünk tovább az ösvényen (GO TRAIL). Ez is zsákutca, de ha itt, a semmi közepén ássunk egy lyukat, és belemászunk (DIG, GO HOLE), egy alagútba kerülünk. Csak keletre indulhatunk (E). Itt megint egy nagy követ látunk, de ezt már tényleg nem tudjuk felvenni. Akkor robbantunk (DROP DYNAMITE, LIGHT FUSE)! Persze érdekesebb kicsit távolabbról kívánni az eredményt, ha egyben akarunk maradni (W, WAIT). Nézzük meg a robbanás eredményét (E): a szikla eltűnt, egy nagy lyuk tátong a helyén (GO HOLE). Egy márvány terembe jutottunk (5. kép). Nyugatra visszamehetünk az alagútba, északra egy lépcső vezet felfelé (N, U) Egy csillámló fátyol hívja magára a figyelmet, de ha megvizsgáljuk (EXAMINE VEIL), megtudjuk, hogy áthatolhatatlan. (N). Egy zeneszobát találtunk a föld alatt! Egy nagy orgonával állunk szemben, ezt érdemes alaposabban szemügyre venni (EXAMINE ORGAN), egy kottát is találunk. Ha már itt vagyunk, muzsikáljunk egy kicsit (PLAY MUSIC). Valami történt, közli a program... A szobában mindenesetre semmi változás, csak visszafele mehetünk (S). Itt viszont rögtön látszik a változás: az „áthatolhatatlan” fátyol eltűnt, egy folyosó nyílt a helyén (GO CORRIDOR). A folyosón mehetünk északra, vagy nyugatra. Ha nyugatra megyünk szembe kerülünk Xerdon-nal, aki egyetlen pillantásával megöl, úgyhogy a 7. képet csak rövidke pillanatig „csodálhatjuk”. Valahogy Xerdon mögé kell kerülnünk, tehát kizárásos alapon az északi folyosón kell mennünk (N, N). Egy ablak előtt állunk, ha kinézünk (LOOK WINDOW) Xerdon-t látjuk. Innen nem tudjuk lenyilazni. Az ablakot nem tudjuk betörni (BREAK WINDOW), csak vandálnak nevez a program. Menjünk vissza délre (S), teljesen logikátlan módon máshová kerülünk, mint amerről jöttünk. Itt egy rést látunk a falon (LOOK SLITS), ahonnan szintén rálátunk a főgonoszra. Már csak egyetlen teendők van (SHOOT XERDON).

dBase II. 2.43 (IS-DOS) – III. rész

3. A dBase II használatba vétele

3.1. A dBASE II indítása

A program CP/M operációs rendszer alatt működik. Ez az ENTERPRISE felhasználóknak azt jelenti, hogy a CP/M-el kompatibilis IS-DOS elindítása után tudjuk a programot használni. A dBase működéséhez két file mindenképpen szükséges: DBASE.COM, DBASEOVR.COM, ezen kívül érdemes használni a help-file-t is: DBASEMSG.TXT. A program indítása a

DBASE

Utasítással lehetséges.

```

IS-DOS
Enter today's date, or return for none
(DD/MM/YY) :21/06/03
Copyright (C) 1982 RSP Inc.
*** dBASE II Ver 2.4 1 April, 1983
Type 'HELP', 'HELP dBASE', or a command
. |

```

A dBASE bejelentkezésekor egy pontot látunk a képernyő alján, ez a dBASE „parancsra várakozásának jele” (promptja), e mögé gépeljük parancsainkat. Ezt a sort nevezzük parancssornak. Mielőtt bármit csinálnánk adatbázis-kezelőnkkel, jegyezzünk meg, néhány jó tanácsot. A program használata közben mindig nézzünk a képernyőre, hátha megjelent rajta egy nekünk szóló üzenet. Elhagyni a programot csak és kizárólag az erre szolgáló

QUIT
paranccsal szabad.

Adataink érdekében ezt az utolsó szabályt soha ne szegjük meg, nagy, árat fizethetünk érte, különösen, ha olyan állomány marad lezáratlanul, amelyben előzőleg módosítottunk valamit.

Adatbázis-állományainkat más parancsokkal is lezárhatjuk, egyet közülük már most megemlítünk, mert ez majdnem tökéletesen alaphelyzetbe állítja a dBASE-t. E parancs formája:

CLEAR

Ez a parancs lezárja az összes adatbázis- és hozzá tartozó segédállományt, visszaáll az egyes számú munkaterületre, és megszünteti az összes létező memóriaváltozót. A dBASE II-nek vannak ún. környezeti paraméterei, melyek különböző használati, működési módokat tesznek lehetővé. Vannak köztük ún. környezeti kapcsolók, me-

lyeknek kétféle «állása» lehet, és vannak olyanok, amelyek egy felhasználó által pontosított értéket állítanak be (szókás «set» parancsoknak is nevezni ezeket, mert a megfelelő beállító parancs minden esetben a SET kulcsszóval kezdődik). Ezek megfelelő kapcsolgatásával, beállításával elérhetjük, hogy a feladathoz, munkastílusunkhoz legjobban illeszkedő környezetet alakítsunk ki magunknak. A dBASE II adatbázis-kezelőt kétféleképpen lehet használni:

A dBASE-t használhatjuk interpreterként, ilyenkor a begépelni parancsaink azonnal értelmeződnek, és amennyiben lehetséges, végrehajtnak. Hibáinkra az adatbázis-kezelő azonnal reagál a hibaüzenetek megjelenítésével. Ha már készítettünk magunknak (vagy más készített nekünk) egy programot a dBASE segítségével, akkor ezen a programot keresztül is használhatjuk az adatbázis-kezelőt. Ilyenkor már a futtatott programon múlik, mennyire látszik, hogy valójában a dBASE-el dolgozunk.

3.2. A dBASE II parancsnyelve

Amikor interpreterként (párbeszédéses módban) használjuk a dBASE-t, parancsainkat közvetlenül a parancssorba gépeljük be. Ilyenkor bizonyos szabályokat be kell tartanunk annak érdekében, hogy utasításainkat a gép is megértse és értelmes módon végrehajtsa. Rendelkezünk ugyanakkor néhány olyan lehetőséggel, amelyek a mi kényelmünket szolgálják. Minden parancsot külön kell begépelnünk és befejezésül az 'ENTER' billentyűt megnyomnunk. Nincs lehetőség arra, hogy egyszerre - egy sorban - több utasítást adjunk ki!

3.2.1. A parancskiadás szabályai

Minden utasításnak parancsszóval kell kezdődnie - általában ez az alapparancs - és a szintaktikai leírásban található formának meg kell felelnie (ld. függelék), csak a megadott paraméterek szerepeltethetők benne.

A parancsszót követő paraméterek, opciók tetszőleges sorrendben gépelhetők be, de a paramétert bevezető kulcsszó nem szakítható el a hozzá tartozó kifejezéstől. (Pl. a „TO <állománynév>” parancsrészletben szereplő két szó között csak szóközők szerepelhetnek.)

Egy parancs legfeljebb 254 karakterből állhat, ha túllépjük a képernyő szélességét, a dBASE a következő sorban folytatja a parancsot, de ez ne zavarjon bennünket. Az 'ENTER' billentyűt csak a végrehajtás kezdeményezésére nyomjuk meg!

A kulcsszavakat, paramétereket egymástól egy vagy több szóközzel kell elválasztani. (Ezek a szóközők is beleszámítanak a 254 karakterbe.)

A kulcsszavak és függvénynevek (a szintaktikai leírásban ezeket csupa nagybetűvel írtuk) rövidíthetők az első négy karakterükkel. (Pl. DISPLAY STRUCTURE helyett DISP STRU írható.) Ha további betűket írunk ki, akkor azoknak is helyesek kell lenniük.

A kulcsszavak, memóriaváltozó-, függvény-, állomány- és mezőnevek kis- és nagybetűkkel egyaránt, de vegyesen is írhatók (pl. DISPLAY = Display = dispLAY).

Mint korábban már említettük, a dBASE II-nek nincsenek fenntartott szavai, a kulcsszavak bármilyen objektum (állomány vagy változó) elnevezésére használhatók, csak hogy ezekkel később nagyon nehezen vágy egyáltalán nem tudunk boldogulni.

A 8 funkcióbillentyűnek alapértelmezésben nincs funkciója, de ha a program indítása előtt definiáljuk őket, használhatjuk a dBase-ben. (A funkcióbillentyűk definiálását célszerű egy BAT-file-ban elvégezni, az EPDOS FKEY parancsa segítségével.) Megjegyzem, a 8. Funkcióbillentyűt ne használjuk, mert a dBase nem szereti.

Programíráskor az ún. páros parancsok egyik fele sem hagyható el, és meghatározott sorrendben, illetve formában használandók (ilyenek: DO WHILE - ENDDO; DO CASE - ENDCASE; IF - ENDIF; TEXT - ENDTXT).

3.2.2. Szintaktika

A parancsok felépítését, formai szabályait nevezzük szintaktikának. Párbeszédés használatkor ezeket mindig szem előtt kell tartanunk, ellenkező esetben rosszul vagy egyáltalán nem tudja végrehajtani utasításainkat a gép. A parancsok általános szerkezete (amelyek tanulmányozásával képeket kapunk a dBASE nyelvi logikájáról, és későbbi szóhasználatunkat is tisztázhatjuk) a következő:

PARANCSSZÓ [PARAMÉTERSZÓ <objektum>

A PARANCSSZÓ írja elő a tevékenységet, melyet el akarunk végezni, a paraméterek mondják meg, hogy mivel és hogyan kell elvégezni. Minden paraméterhez tartozik egy kulcsszó (PARAMÉTERSZÓ), de sokszor ez elhagyható. A parancsok leírásában használt jelöléseket egy példa segítségével mutatjuk be (magáról a parancsról majd később lesz szó).

Nézzük például a következő parancsot:

USE <állománynév> [INDEX <indexállománynév>]

Jelölések:

A [...] közti paraméterek, parancsrészletek elhagyhatók, használatuk nem kötelező.

A <...> közé írt szöveget a felhasználónak kell a megfelelő tartalommal pontosan megadni (pl. „<állománynév>” helyett „karton” írható). A [, < > jelek nem részei a parancsoknak, ezeket begépelni nem szabad. (Figyelem: parancsokban a „[„ és a „]” jelet karaktersorozat határolójeleként értelmezi a rendszer!)

A különböző „listák” elemeit vesszővel (,) kell egymástól elválasztani, vessző másutt nem szerepelhet.

A NAGYbetűs szavak pontosan úgy írandók (eltekintve a rövidítéstől), ahogyan a leírásban látható

3.2.3. A rekordkezelő parancsok szerkezete

A parancsok egy jelentős része az előírt tevékenységet kifejezetten az adatbázis rekordjaival végzi (pl. megjele-

nít, másol, töröl stb.) ezeket hívjuk rekordkezelő parancsoknak. Általános szerkezetük:

PARANCSSZÓ [<érvényességi kör>] [WHILE <feltétel>] [FOR <feltétel>]

Nézzük végig az egyes paramétereket!

Az érvényességi kör (scope)

Az adatbázisnak azt a részét határozza meg, amelyen a parancsot végre kell hajtani. Az alapértelmezés általában az „aktuális rekord” néhány parancsnál „ALL” (= minden), azaz a teljes adatbázis. (A FOR és WHILE paraméterek megváltoztatják az alapértelmezést; mellettük az „ALL” érvényességi kör megjelölés használata felesleges, az nem változtatja meg a parancsban részt vevő rekordok körét.) Az „<érvényességi kör>” lehet:

RECORD <n>	az n. rekordon hajtja végre a parancsot.
NEXT <n>	az aktuálistól kezdve a darab rekordon (beleértve a aktuálisat is).
ALL	az adatbázis-állomány összes elérhető rekordján végrehajtja a parancsot.

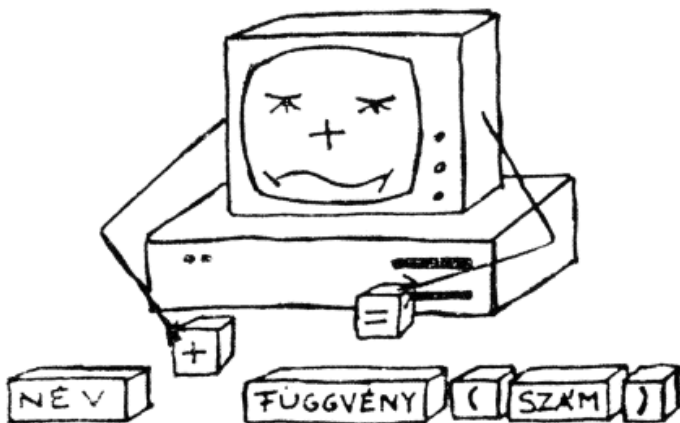
Ha az alapértelmezés az „aktuális rekord”, és más érvényességi kört nem írunk ki, a parancs azon a rekordon hajtódik végre, amelyre a rekord mutató („record pointer”) mutat. Minden megnyitott adatbázisunkhoz tartozik egy rekordmutató, mely közvetlenül megnyitás után a logikai sorrendben első rekordra mutat. Bizonyos parancsok közvetlenül állítják a rekordmutatót egy adott rekordra, mások közvetve, a parancsvégrehajtás, közben mozdítják el korábbi helyéről. Általában elmondható, hogy a rekordmutató azt a rekordot címzi meg, amelyekkel az adatbázis-kezelő utoljára foglalkozott. Az „ALL” érvényességi körű parancsok végrehajtása után a rekordmutató az utolsó rekord utáni, nemlétező rekordot (annak sorszámát) jelöli ki.

Kifejezés (expression)

A kifejezés különböző műveleti jelekkel (operátorokkal) összekapcsolt, vagy a megengedett függvények argumentumában elhelyezett állandókból és változókból áll. Az operátorok két oldalán azonos típusú adatnak kell szerepelnie. Különböző típusok esetén konverziós függvényeket használhatunk. A kifejezés a kiértékeléskor kapott eredménytől függően lehet karakteres (erre a továbbiakban „Kkif” rövidítést használunk), numerikus („Nkif”) és logikai („Lkif”) típusú.

Feltétel (condition)

A feltétel olyan kifejezés, amelynek értéke logikai típusú (általában összehasonlító vagy logikai műveletek eredménye, vagy logikai értéket szolgáltató függvényhívás).



A WHILE és FOR paraméter

Hasonlóan az érvényességi körhöz, ez a két paraméter is a parancsban részt vevő rekordok körét változtatja meg. Ennél a két paraméternél azonban nem a rekord állományon belül elfoglalt helye, hanem tartalmának tulajdonságai töltik be a döntő szerepet. Minden olyan parancsnak paramétere, amelynél érvényességi kör is megadható.

A WHILE paraméter a rekordmutató aktuális pozíciójától kezd vizsgálni a rekordokat, magát a parancsot csak akkor hajtja végre, ha a feltétel igaznak bizonyul. Az első olyan rekordon, amelyre a feltétel nem teljesül, befejeződik a parancs végrehajtása. A paraméter működéséből következik, hogy „ALL” vagy „REST” érvényességi kör használata ilyenkor felesleges, nem változtat a parancsvégrehajtás mikéntjén „NEXT” érvényességi körrel lehetséges, hogy nem a feltétel hamissá válása, hanem az érvényességi kör vége állítja meg a parancs végrehajtását.

A FOR paraméter az érvényességi kör alapértelmezését átírja „ALL” értékre, tehát a végrehajtás a rekordmutató pillanatnyi tartalmától függetlenül mindig a logikailag legelső rekordon kezdődik, és az adatbázis végéig tart. A parancsban előírt műveletet a dBASE mindazokon a rekordokon elvégzi, amelyekre a feltétel kiértékelése „igaz” értéket szolgáltat. FOR paraméter mellett a „REST”, vala-

mint a „NEXT érvényességi kör használata befolyásolhatja, hogy mely rekordokon értékelődik ki a feltétel.

„RECORD <n>” érvényességi kör mindkét paraméter mellett ugyanazt eredményezi: ha a feltétel „igaz”, a parancs végrehajtódik az adott sorszámú rekordon. A két paraméter együtt is használható. Ekkor a WHILE-nak van prioritása, azaz a WHILE feltétele határozza meg a végrehajtás befejezésének pillanatát. Használható például olyan szituációkban, amikor az adatbázis egy adott részletén (amiről nem tudjuk, hogy hány darabból áll, csak azt, hogy egy adott feltételnek eleget tevő rekordokat tartalmaz) további feltételtől függően szeretnénk egy parancsot végrehajtani. (Ezt elérhetnénk másképpen is, pl. FOR paraméter használatával és összetett, logikai műveleteket tartalmazó feltétel megfogalmazásával, de ekkor minden rekordon végig kell mennie az adatbázis-kezelőnek, és ez bizony időbe telik.)

3.3. A munkaterületek használata

A dBASE II egyszerre 2 nyitott adatbázis-állományt tud kezelni. Ezek egymástól függetlenek, mindkettőn egy adatbázis-állomány és a hozzá tartozó egyéb típusú állományok tarthatók nyitva. Mindkét munkaterületen önálló, alapértelmezés szerint a másiktól független rekordmutató van, segítségükkel két munkaterület között később logikai kapcsolatot tudunk létesíteni. A dBASE elindításakor az első munkaterület az aktuális. A két munkaterület neve PRIMARY (az első), SECONDARY (második). A munkaterületek között váltani s SELECT paranccsal tudunk:

```
SELECT PRIMARY
SELECT SECONDARY
```

A kiadott parancsok túlnyomó többsége arra a munkaterületre vonatkozik, amelyen kiadtuk. A memóriaváltozók nincsenek munkaterülethez kötve, azokat bárhonnán elérhetjük és módosíthatjuk.

Folytatjuk!

A 2018-as év „termése”

Játékok:

Chibi Akumas, Operation Alexandra, Profanation, Crillion, Quadrillion, Banana, Banana Plus, RayTrace, Life and Death

Util:

MOD Player

Zenék:

Néhány (sok) midi zene

Demo:

Nyan Cat (+ télapós különkiadás)

Hardver:

EPower

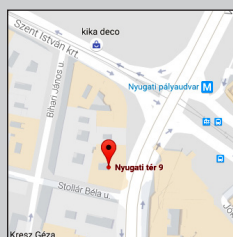
Enterprise Klub 2018. október 20.

ENTERPRISE

KLUB

Egy évben 6 alkalommal

Helyszín:
Nyugati Oktatási
Központ, Skála terem
Budapest (V. ker.)
Nyugati tér 9.
14 órától 19 óráig



További információ: www.enterpriseklub.hu

Ha te is szeretnél Az ENTERPRESS Magazin szerkesztője lenni, küldj cikket, játékleírást, játékismertetőt, vagy bármit amely az Enterprise számítógéppel kapcsolatos!

A cikkeket erre az e-mail címre küldheted:

info@enterprise.news.hu

ENTERPRISE

FOREVER

<https://enterpriseforever.com>

ENTERPRESS Magazin - 2018/6. november-december

Főszerkesztő: Matusa István

Szerkesztőségi főmunkatárs: Németh Zoltán (Zozosoft)

A csapat: geco, Povi, Kiss László, SzörG, szipucsu, lgb, Hajdó Máté †

Design, nyomdai előkészítés: Matusa István

Weboldal: <http://enterprise.news.hu>

E-mail: info@enterprise.news.hu

A lap időszakosan - korlátozott példányszámban - nyomtatott formátumban és elektronikus formában is megjelenik.

ENTERPRESS e-magazinok:

<http://enterprise.news.hu/index.php/magazin>