

MOJE Atari

CENA 8000 ZŁ

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW ATARI • WRZESIEŃ/PAŹDZIERNIK (5'91)

7



WIEŻA ST

**KASSETOWA BAZA DANYCH
PROGRAMOWANIE TOMS
NOWY REKORD GUINNESSA**

CO? W n u m e r z e:

Drogi Czytelniku!

Świat komputerów zmienia się szybko jak w kalejdoskopie i niekiedy trudno nawet nadążyć. Najważniejszym elementem tych przemian jest dla nas niemal całkowite zniknięcie z zachodnich rynków komputerów ośmiobitowych. Nawet szesnastobitowe — jeszcze kilka lat temu uważane za super nowoczesne — są wypierane przez 32-bitowe, a już mówi się o następnej generacji.

W świecie Atari obserwuje się brak zainteresowania modelami XL i XE, których miejsce — jako komputerów domowych: do grania i prostszych prac — zajęły teraz ST i STE. Na ich miejsce wszedł z kolei Atari TT — komputer do profesjonalnych prac głównie z dziedziny DTP i grafiki. Bliżej opiszemy te zjawiska w następnym numerze w relacji z targów „Atari Messe”, które odbyły się pod koniec sierpnia w Düsseldorfie.

Na naszym podwórku zmiany te są jednak jeszcze odległe. Trudno mówić bowiem o ST i TT, gdy wielu fanów komputeryzacji nie może sobie pozwolić nawet na zakup stacji dysków do „małego” Atari. Mimo wszystko zmiany takie będą zachodziły, choćby tylko ze względu na zmieniające się relacje cenowe. Na przykład, za cenę niewiele większą niż zestawu Atari XE ze stacją dysków (około 4 mln zł) można już nabyć IBM PC/XT w najprostszej konfiguracji. Nie dorasta on wprawdzie nawet do pięt komputerom ST (zapytaj OBIEKTYWNEGO fachowca), lecz jest to przecież następna generacja. Wszystko to będzie zmuszało nas do poświęcania coraz większej części naszego pisma komputerom ST i TT, aż do całkowitej rezygnacji (za kilka lat) z materiałów poświęconych komputerom ośmiobitowym.

O czym lojalnie informuje

Wojciech ZIENTARA

PROGRAMOWANIE

Programowanie stacji TOMS Multi 15
Użytkownicy stacji TOMS Multi mogą programować ją samodzielnie. O tym, w jaki sposób to zrealizować, pisze autor systemu operacyjnego stacji.

Tylko dla zaawansowanych (4) 12
W poradach dla programistów tym razem o zabezpieczaniu przed kopiowaniem programów zapisanych na kasetach magnetofonowych.

PEEK, POKE i nie tylko 13
Dodatkowe procedury i funkcje dla piszących programy w Kyan Pascalu 2.0.

GRY

Starcie 6
I znowu brzydki kosmici napadają na Ziemię. Bądź dzielny, zręczny i sprytny — obroń naszą planetę przed najazdem. W tobie ostatnia nadzieja.

Skoczek 6
Nowa gra komnatowa oparta na motywach „Robbo”. Niestety, tylko dla nabywców naszej dyskietki z programami.

Biblioteka rozdań brydżowych 7
Program dla zapalonych brydżystów — możliwość zapisywania dowolnej liczby rozdań ze znanej gry „Blue Team Bridge”.

Nowe i stare 23
Opisy gier „Gato”, „Archon”, „Gunship” i „Starglider”.

Nowy rekord Guinnessa 25
Jak długo można grać bez przerwy w jedną grę komputerową? Przeczytaj, to się przekonasz.

TOP 12 22

PROGRAMY UŻYTKOWE

Kasetowa baza danych 9
Nie jest to wprawdzie „SynFile”, lecz z pewnością przyda się użytkownikom magnetofonów do przechowywania swoich danych.

Ramdisk 130XE 10
Atari 130XE i magnetofon, to zestaw często spotykany. Dodatkowa pamięć komputera może być w takim zestawie wykorzystana do ułatwienia pracy.

SPRZĘT

Systemy instalowane w stacjach dysków 14
Charakterystyka instalowanych w stacji Atari 1050 systemów: Speedy 1050, Happy Warp 7.10 i I.S. Plate.

SZKOŁKA

Ruch statku 11
Najdłuższą procedurą gry „Konwój” jest Ruch Statku. Jej zaprojektowanie rozpoczyna etap tworzenia właściwej gry.

Wprowadzanie danych 21
Co piszczy w Basicu? Różne różności. Na początek przeczytaj garść informacji o wprowadzaniu danych do programu.

INSTRUKCJE

Phantasie 19
Kolejną pozycją „Vademecum Atari” jest jedna z najlepszych gier *role-playing* — „Fantazja”, czyli czarownicy, wampiry, duchy i drużyna dzielnych bohaterów.

DZIAŁ ST

1st Word i drukarki 28
Koniec kłopotów z polskimi literami na drukarce — niezależnie od jej typu, czyli jak nauczyć „1st Word Plus” drukowania po polsku.

Outline Art 29
Podręczny wykaz funkcji programu umożliwiającego tworzenie grafiki wektorowej do „Calamusa” i nie tylko.

Mysz z joysticka 31
Joystick bywa wygodniejszy od myszy. Prosta przystawka umożliwi podłączenie go do komputera tak, aby nie zauważył on zmiany.

Wieża — nie tylko figura szachowa 32
Komputery Atari to nie tylko firma Atari. Inne firmy także oferują ten sprzęt, lecz w odmiennej postaci.

INNE

Lynx 4
Nowa maszynka do grania, czyli druga generacja automatów z rodziny „Video Game” — godny następcy Atari 2600.

Wielki konkurs 25

Więści 3

Listy 4

Edytor Basica 5

Wpisywanie znaków specjalnych 5

UWAGA: Wszystkie programy opublikowane w naszych wydawnictwach o Atari („Tylko o Atari” 1 i 2 oraz „Moje Atari” 1–7) można nabyć na dyskietkach dostępnych w cenie 15 000 zł w redakcji: Warszawa, ul. Wasilkowskiego 7, wtorek–czwartek, godz. 10–15. Ponadto dyskietki te są sprzedawane przez sklepy „Bajtki” w Krakowie i Bytomiu oraz niektóre inne sklepy komputerowe.

MOJE Atari

Redaktor naczelny:
Wojciech Zientara
Sekretarz redakcji:
Robert Chojecki

Opracowanie graficzne:
Wanda Roszkowska
Redaktor techniczny:
Maria Radziwińska
Zdjęcia:
Leopold Dzikowski

Redakcja:

Spółdzielnia „Bajtek”
00-687 Warszawa, ul. Wspólna 61
tel. 21-12-05

Skład i druk:

Przedsiębiorstwo Wydawniczo-Poligraficzne „GRYF”
Ciechanów, ul. Sienkiewicza 51
Fotoskład:
Wiesława Macielak
Montaż offsetowy:
Grażyna Ostaszewska
Korekta:
Maria Krajewska
Teresa Rutkowska
Nr zlecenia: 63441
Nakład: 105 tys. egz.
ISSN 0867-390X

Redakcja zastrzega sobie prawo skracania i adnotacji nadestanych artykułów. Materiałów nie zamówionych nie zwracamy. Za treść ogłoszeń redakcja nie odpowiada.



Lynx — konsola do gier video

LYNX

Produkcja gier i komputerowych urządzeń dla graczy stanowią dziś pokaźną gałąź przemysłu elektronicznego. Dla milionów użytkowników gry video są praktycznie jedynym rodzajem programów uruchamianych na komputerze. Nic dziwnego, że i firma Atari prowadzi prace nad konstrukcją elektronicznych urządzeń przeznaczonych wyłącznie dla graczy.

Jednym z nich jest Lynx, przenośna konsola do gier video. Lynx ma niewielkie wymiary, zaokrąglone kształty i łatwo mieści się w dłoni. Można go włożyć do kieszeni i pobawić się nim podczas podróży pociągiem, samochodem, albo w przerwie lekcyjnej.

Na miniaturowym ciekłokrystalicznym ekranie komputerka jest generowany obraz w 4096 różnych odcieniach kolorów. Wyświetlacz jest bardzo delikatnym, a przy tym prawdopodobnie najdroższym elementem całego urządzenia, dlatego chroni go przezroczysta płytka z tworzywa sztucznego. Porysowaną lub uszkodzoną szybę można bardzo łatwo wymienić na nową. We wnętrzu Lynxa jest wbudowany głośnik, przez co osiągnięto bardzo wysoką, jak na tak niewielkie

urządzenie, jakość dźwięku. Jasność obrazu i siła głosu jest regulowana. Leworęczni gracze mają takie same szanse jak praworęczni, ponieważ po odwróceniu konsoli można zmienić orientację ekranu. Naciśnięcie odpowiedniej kombinacji klawiszy powoduje odwrócenie obrazu o 180°. Zamiast joysticka zastosowano plastikowy manipulator w kształcie krzyża, który może być pochylony w 8 kierunkach.

Gry dostarczane są na specjalnych kartach, które mają wymiary 57x60 mm i 2 mm grubości. Przyłącza się je do portu ROM. Kart nie wolno wymieniać podczas pracy komputera, gdy jest włączone zasilanie. Przy pomocy specjalnego kabla można połączyć 8 Lynxów. Tym samym gracze mogą przeżywać dodatkowe emocje wynikające z rywalizacji z kilkoma innymi partnerami. Użytkownik Lynxa na pewno nie będzie się nudzić, ponieważ opracowano już sporą liczbę gier przystosowanych do tego urządzenia. Niektóre z nich są wersjami znanych szlagierów z ST, Amigi lub C-64. Wśród wielu tytułów znajdziemy: Miss Pac Man, Gauntlet, Gates of Zendocon, Robo Squash, Electrocop, Paperboy, Blue Lightning, Klax. Gwarantuje one znakomitą zabawę na długie miesiące.

Komputer zbudowano na bazie 8-bitowego procesora 6502, który taktowany jest z niezwykle dużą szybkością 16 MHz. CPU jest zamontowany w specjalnej podstawie wewnątrz komputerka. Pokrywę Lynxa wykonano z tworzywa sztucznego i od wewnątrz pokryto warstwą metaliczną, która zastępuje to metalowe ekrany chroniące przed zakłóceniami elektromagnetycznymi.

Lynx jest zasilany z baterii lub z sieci. Zasilacz sieciowy stanowi wyposażenie standardowe. Baterie umożliwiają prowadzenie gry przeciętnie przez około 2 godziny. Wyczerpywanie się baterii objawia się zmniejszeniem intensywności barw na ekranie i osłabieniem siły głosu głośniczka.

Lynx kosztuje w RFN ok. 399 DM, a karty z grami po ok. 70 DM.

Janusz JARMOCH

MONITOR SC1435

Na początku tego roku zaprezentowano Mega STE z kolorowym monitorem, oznaczonym symbolem SC 1435. Nowy monitor Atari od razu zdobył przychylne opinie specjalistów. Może on pracować z każdym komputerem, który ma analogowe wyjście RGB. Dzięki temu nadaje się do wszystkich komputerów ST.

Monitor zasługuje na uwagę nie tylko ze względu na estetyczny, nowoczesny wygląd. Jakość obrazu SC 1435 znacznie przewyższa efekty uzyskiwane w tradycyjnym SM 124. Dużą

zaletą jest także dobrej jakości, stereofoniczny dźwięk.

Większość elementów obsługowych monitora znajduje się w jego przedniej części, tuż pod ekranem. Są to pokrętła regulacji kontrastu, jasności, siły głosu, synchronizacja pionowej oraz przełącznik trybu monochromatycznego, umożliwia przełączenie na tryb emulacji zielonego monitora monochromatycznego. Jedynym krytycznie ocenianym elementem jest wyłącznik sieciowy, który umieszczono z tyłu. Używa się go bardzo często, a więc ze względu na wygodę obsługi powinien być umieszczony w łatwo dostępnym miejscu.



Atari Mega ST
z kolorowym
monitorem SC 1435
i drukarką
laserową SLM 605

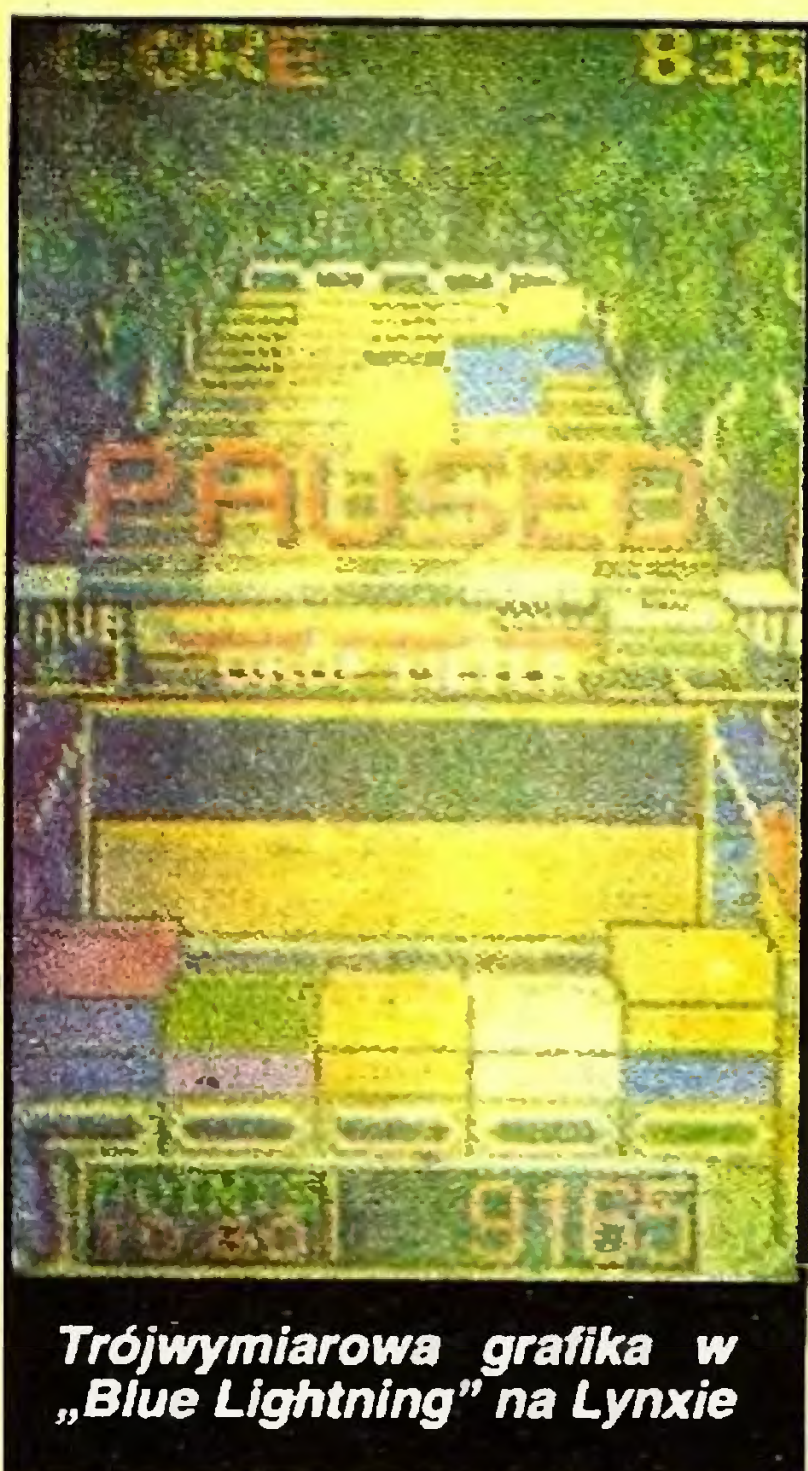
ST de LUXE

Od niedawna ukazuje się w RFN magazyn „ST de Luxe”, przeznaczony dla fanów Atari ST.

Czasopismo to jest wydawane w nietypowej, ale bardzo dogodnej dla użytkownika komputera formie. Wszystkie informacje publikowane są na dyskietkach. Dyskowy magazyn zajmuje się przede wszystkim popularyzacją najciekawszych informacji o oprogramowaniu. Taka nowoczesna forma przekazu ma ogromną przewagę nad tradycyjnymi artykułami drukowanymi na papierze. Na dyskietkach znajdują się programy demonstracyjne, symulacje, gry, porady dla programistów. Wszystko można natychmiast obejrzeć i wypróbować na komputerze. Takiego praktycznego testu nie zastąpi nawet najlepszy opis programu, ilustrowany



pięknymi, kolorowymi zdjęciami. Magazyn na dwóch dyskietkach kosztuje 19,80 DM. Lektura pisma wydawanego na dyskietkach jest dużą atrakcją dla każdego właściciela Atari ST.



Trójwymiarowa grafika w „Blue Lightning” na Lynxie

L I S T Y

Czy na emulatorze AT-Speed można uruchamiać programy z komputera XT? Czy do Atari ST można podłączyć tylko dwie stacje dysków? Czy można podłączyć Atari 1040STE do monitora Neptun 156?

**Rafał Lenczewski
Żyrardów**

AT-Speed jest emulatorem komputera IBM PC/AT, na którym działają niemal wszystkie programy dla PC/XT. Nie ma więc powodu, dla którego programy te nie mogłyby również działać na AT-Speed. System operacyjny Atari ST umożliwia rozpoznanie dwóch stacji dysków. Stosowanie większej ich liczby jest niecelowe, gdyż nie istnieje taka potrzeba. Jeśli jednak dwie stacje dysków są niewystarczające, jest to sygnałem, że należy nabyć twardy dysk. Podłączenie monitora Neptun 156 jest możliwe, lecz tylko w niektórych wersjach ST, a ponadto będzie on pracował jedynie w niskiej i średniej rozdzielczości. Odpowiednie sygnały należy pobrać z gniazda monitorowego: styk 1 — fonia, styk 2 — wizja i styk 13 — masa.

W „Bajtku” 9—10/90 znalazłem ogłoszenie firmy z Lublina. Kupiłem tam dyskietki „A PLUS” *WARE” (przesyłam jedną). Dyskietki te nie formatują się w mojej stacji dysków Atari 1050. Formatują się jednak na komputerach klasy IBM w stacjach 1.2 MB. Nie wiem, co zrobić z tymi dyskietkami, aby osiągnąć format odpowiedni dla stacji 1050.

**Wojciech Sterna
Poznań**

Nadesłana dyskietka jest przeznaczona do formatowania w wysokiej gęstości (HD = High Density). Niektóre dyskietki tego typu mają specjalny rodzaj nośnika magnetycznego, co uniemożliwia ich sformatowanie w mniejszej gęstości. W zasadzie jedynym rozwiązaniem jest w takim przypadku zamiana dyskietek na „zwykłe”, czyli co najwyżej podwójnej gęstości (DD). Niekoniecznie trzeba tą sprawę

załatwiać z firmą w której dokonano zakupu — z pewnością znajdą się chętni do zrealizowania takiej zamiany wśród znajomych lub w jakiejś firmie dysponującej komputerami IBM.

*

Jesteśmy pacjentami Wojewódzkiego Szpitala Chirurgii Urazowej w Piekarach Śląskich. Z nudów, jakie tu panują, postanowiliśmy rozwiązać wszystkie zagadki kryminalne w grze „Baker Street 221B”. Podajemy więc gotowe cyfry, których należy użyć przy rozwiązywaniu poszczególnych zagadek. Jeśli ktoś chce dojść do tego samodzielnie, to dalej nie powinien czytać.

- | | |
|----------------------------------|---------------|
| 1. dziwny facet | — 3, 2, 1; |
| 2. srebrna plamka | — 3, 1, 3; |
| 3. zemsta kameleona | — 4, 2, 2; |
| 4. tajna wiadomość | — 1; |
| 5. zejście urzędnika | — 1, 4, 3; |
| 6. zapowiedziana śmierć | — 3, 2; |
| 7. ograbiony właściciel lombardu | — 2, 1, 2; |
| 8. puste ręce złodziejki | — 4, 2, 3; |
| 9. dziwna sprzątaczką | — 4, 2, 1; |
| 10. zemsta księżniczki | — 3, 2, 1; |
| 11. niezwywy gość | — 4, 1, 4; |
| 12. woła starej panny | — 2, 2, 2; |
| 13. nieznoszona panna młoda | — 1, 1, 1; |
| 14. zakochany żeglarz | — 2, 4, 1; |
| 15. nieznaną nieboszczyk | — 4, 2, 1; |
| 16. porwana piosenkarka | — 2, 1, 2; |
| 17. otruty pokerzysta | — 1, 2, 3; |
| 18. żarłoczna plotkara | — 4, 1, 4; |
| 19. przewidujący złodziej | — 4, 4; |
| 20. łagodny poborca podatkowy | — 2, 2, 2; |
| 21. upadły anioł | — 4, 1, 2; |
| 22. szpieg | — 4, 2, 1; |
| 23. tajemnicza czaszka | — 2, 3, 1; |
| 24. muzyczny morderca | — 4, 2, 4; |
| 25. oko tygrysa | — 4, 1, 2; |
| 26. zagadkowe zabójstwo | — 1, 3, 4; |
| 27. wyzwanie Moriarty'ego | — 2, 4, 2; |
| 28. ostatnia skarga doktora | — 1, 3, 2; |
| 29. tajemnica morderstwa | — 3, 3, 2, 1; |
| 30. zamordowany makler | — 4, 1, 2. |

**Wojciech Krytowski
Dawid Kacperak**

TOMS wita Was! Nowości firmy TOMS

W niniejszym odcinku ogłoszeniowym polecamy: Bardzo tanie stacje 5.25" dla ST i Amigi w wielu odmianach, z dodatkowymi funkcjami. Takich stacji nie proponuje nikt inny!

Samplery dla ST i Amigi. Sampler ST posiada oryginalne polskie oprogramowanie.

Ponadto informujemy, że właśnie został opracowany interfejs RS232 przeznaczony do dobudowania do stacji TOMS 720 (test w MA 5/91). W związku z tym oferujemy i taką wersję naszej stacji!

Oczywiście polecamy także wszystkie wcześniejsze nasze opracowania, jak oryginalne rozszerzenia pamięci, usprawnienia stacji dysków itd, opisywanie w Bajtkach i Moim ATARI.

Nasza firma gwarantuje solidność, niezawodność i wysoki poziom techniczny.

**Nasz adres: TOMS, ul. Widok 14/1,
00-023 Warszawa
Tel. (0-22) 27-16-01, (0-2) 641-54-29
Zapraszamy w godzinach 9-15!**

Atari Turbo 2000 F

Nowy system transmisji danych z magnetofonem przyspieszony do 6700 bodów.

Komplet:

- cartridge
- oprogramowanie
- przeróbka magnetofonu
- instrukcja obsługi
- 12 miesięcy gwarancji

Instalacje wykonujemy na poczekaniu.

Interfejs do zwykłego magnetofonu.

Duży wybór oprogramowania w standardzie TURBO-2000.

Informacja: Tel. 33-40-91

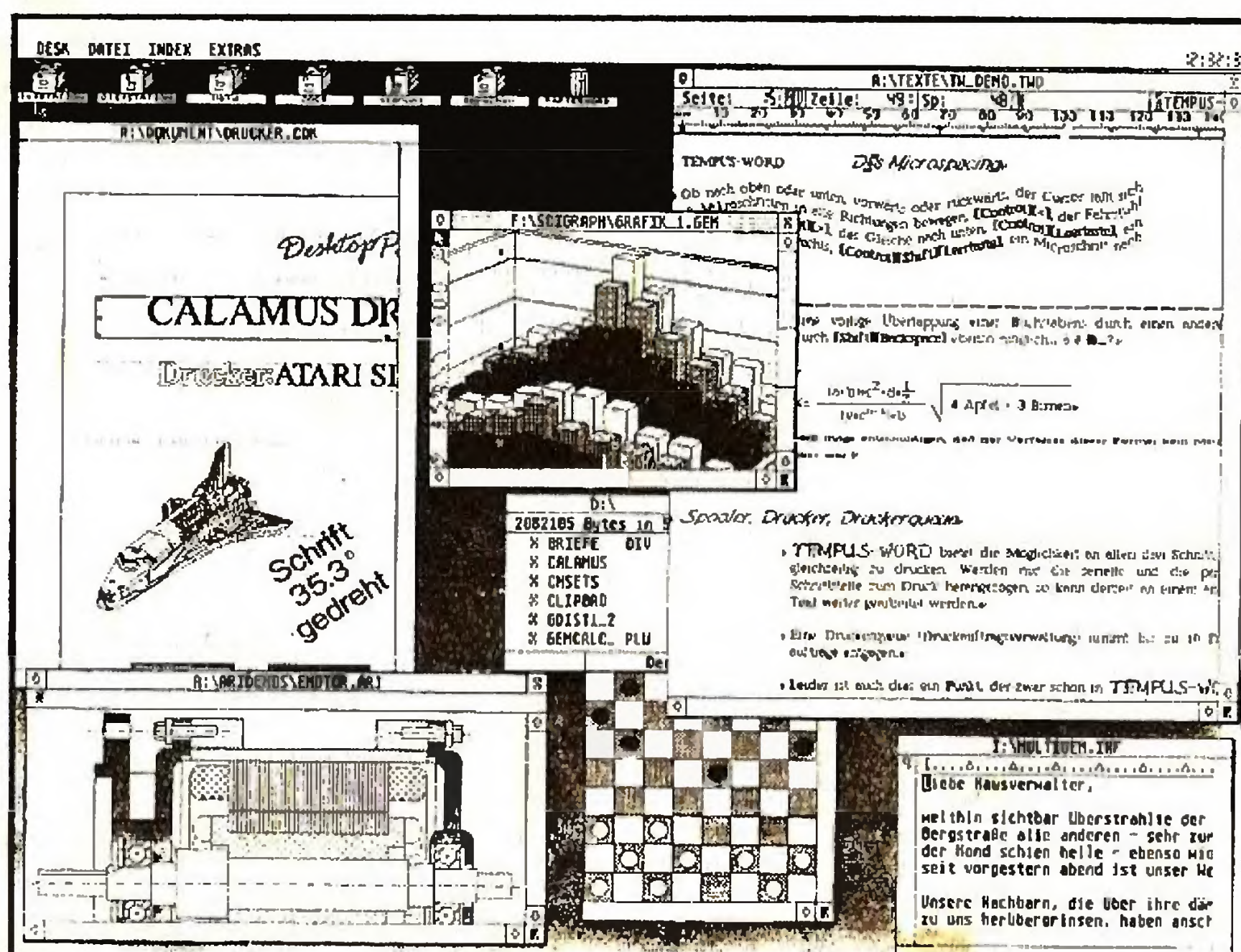
Korespondencja:

MUEL

**Ul. Cząstkowska 30
01-678 Warszawa**

B82

Wygląd ekranu w wielozadaniowym rozszerzeniu MultiGEM do systemu operacyjnego Atari ST.



MultiGEM

Użytkownicy Atari ST mają wreszcie do dyspozycji wielozadaniowe rozszerzenie systemu operacyjnego. Wielozadaniowy system MultiGEM był prezentowany na targach CeBIT'91 przez firmę MAXON Computer GmbH. Umożliwia on równoległą realizację 6 zadań. Każdy z programów ma przydzielone oddzielne okno. Uaktywnienie programu następuje po najechaniu myszą i „tupnięciu” w odpowiadającym mu oknie. Programy, aktualnie nieaktywne nadal pracują w tle i mogą być w każdym momencie wywołane. Specjalny tryb pracy powoduje

przełączenie programów kolidujących z MultiGEM do normalnego GEM Atari.

MultiGEM nadaje się do zastosowania w każdym Atari ST z systemem operacyjnym TOS od wersji 1.2. System współpracuje z monitorem monochromatycznym lub kolorowym, z monitorami wielkoformatowymi, a także kartami rozszerzającymi z procesorem 68020 lub 68030, które przyspieszają pracę komputera.

Orientacyjna cena wielozadaniowego systemu MultiGEM do Atari ST wynosi 159 DM. W przyszłości producent zapowiada także opracowanie wersji MultiGEM na Atari TT.

Przy przepisywaniu programów w Basicu nie sposób ustrzec się błędów.

Aby uniknąć żmudnego wyszukiwania popełnianych omyłek wszystkie programy w Basicu są publikowane wraz z kodami kontrolnymi. Zamieszczony obok program służy do kontroli tych kodów podczas przepisywania programu.

Wydruk „Edytora Basica” należy dokładnie przepisać i zapisać na kasecie lub dyskietce (najlepiej instrukcją LIST „C:” lub LIST „D:EDYTOR.LST”). Poprawność przepisania można sprawdzić samym „Edytorem” w opisany niżej sposób.

Przystępując do wpisywania dowolnego programu z naszego pisma trzeba najpierw wczytać i uruchomić „Edytor Basica”. Następnie należy przepisywać kolejne wiersze programu. Po wpisaniu każdego wiersza i naciśnięciu klawisza RETURN wiersz ten pojawia się w dolnej części ekranu wraz z obliczonym kodem kontrolnym. Jeżeli wyświetlony kod jest taki sam, jak wydrukowany przed numerem wiersza, można przystąpić do wpisywania następnego wiersza. Jeśli kody są różne, to ponowne naciśnięcie RETURN wyświetla wpisany wiersz w górnej części ekranu i umożliwia dokonanie w nim niezbędnych poprawek.

Samo naciśnięcie RETURN

wywołuje zawsze ostatnio wpisany wiersz. W celu wywołania innego, wcześniej napisanego wiersza trzeba podać jego numer poprzedzony gwiazdką (np. 1000) i nacisnąć RETURN. Wpisanie samej liczby powoduje wymazanie z pamięci komputera wiersza programu o takim numerze.

Po poprawnym przepisaniu całego programu trzeba przerwać pracę „Edytora Basica” przez naciśnięcie klawisza BREAK lub RESET. Następnie w celu usunięcia „Edytora” i zbędnych zmiennych (z tablic nazw i wartości zmiennych) zapisujemy program na kasecie instrukcją LIST „C:”, 0,31999 lub na dyskietce instrukcją LIST „D:nazwa”, 0,31999. Teraz kasujemy zawartość pamięci komputera instrukcją NEW i odczytujemy program przy pomocy ENTER „C:” (z kasety) lub ENTER „D:nazwa” z dyskietki. Po wykonaniu tych czynności w pamięci komputera znajduje się tylko gotowy program bez „Edytora Basica” i można go już ostatecznie zapisać na odpowiedni nośnik.

Procedura ta jest może nieco kłopotliwa, lecz zabezpiecza w stu procentach przed popełnieniem omyłki przy przepisywaniu programu. PAMIĘTAJ: ZAWSZE UŻYWAJ „EDYTORA BASICA”.

```

NC 32000 REM EDYTOR BASICA
SE 32010 REM wersja 1.1 dla "Bajtki"
JB 32020 CLR :DIM LINIA$(120):CLOSE #2:CL
    OSE #3
MN 32030 OPEN #2,4,0,"E:":OPEN #3,5,0,"E:
    "
ZR 32040 ? CHR$(125):POSITION 11,1:? "ED
    YTOR BASICA"
YB 32050 TRAP 32040:POSITION 2,3:? "Wpisz
    linie programu"
KH 32060 POSITION 1,4:? " ":INPUT #2;LINI
    A$:IF LINIA$="" THEN POSITION 2,4:LIST
    B:GOTO 32060
XL 32070 IF LINIA$(1,1)="" THEN B=VAL(LI
    NIA$(2,LEN(LINIA$)):POSITION 2,4:LIST
    B:GOTO 32060
TH 32080 POSITION 2,10:? "CONT"
BI 32090 B=VAL(LINIA$):POSITION 1,3:? " "
    ;
NY 32100 POKE 842,13:STOP
CN 32110 POKE 842,12
KI 32120 ? CHR$(125):POSITION 11,1:? "ED
    YTOR BASICA":POSITION 2,15:LIST B
BZ 32130 C=0:ODP=C
LL 32140 POSITION 2,16:INPUT #3;LINIA$:IF
    LINIA$="" THEN ? "LINIA ";B;" USUNIET
    A":GOTO 32050
GU 32150 FOR D=1 TO LEN(LINIA$):C=C+1:ODP
    =ODP+(C*ASC(LINIA$(D,D))):NEXT D
SR 32160 KOD=INT(ODP/676)
BX 32170 KOD=ODP-KOD*676
BM 32180 KODS=INT(KOD/26)
VF 32190 KODM=KOD-(KODS*26)+193
NF 32200 KODS=KODS+193
UM 32210 POSITION 0,16:? CHR$(KODS);CHR$(
    KODM)
HN 32220 POSITION 2,13:? "Jeżeli kod sie
    nie zgadza, naciśnij RETURN i popr
    aw linie.":GOTO 32050

```

Część programów publikowana w naszym piśmie jest drukowana w trybie graficznym.

Jest to spowodowane występowaniem w tych programach znaków, które są stosowane tylko w komputerach Atari i nie mogą być przedstawione przez drukarkę w trybie znakowym.

W trybie graficznym znaki znajdujące się obok siebie są

```

ABCDEFGHIJKLMN OPQRSTUVWXYZ
ABCDEFGHIJKLMN OPQRSTUVWXYZ
abc defghi jklm nopqrst uvwxyz
abc defghi jklm nopqrst uvwxyz
0123456789 0123456789
!"#$%&'()*<>-=;+*,./_:\^[]?
!"#$%&'()*<>-=;+*,./_:\^[]?

```

Zestaw znaków Atari zawiera ponadto znaki specjalne, które wymagają specjalnego wpisywania. Wygląd tych znaków oraz klawisze, które służą do ich uzyskania, są pokazane w tabelkach umieszczonych obok. Zawsze, gdy są używane klawisze <CONTROL> (w tabelkach CTRL) i <SHIFT>, trzeba trzymać je wciśnięte podczas naciskania następnego klawisza. Gdy konieczne jest użycie klawisza <ESC>, należy go nacisnąć tylko jeden raz i zwolnić przed naciśnięciem następnego klawisza. Znaki pokazane w tabeli „INVERSE” są negatywami znaków z tabeli „NORMALNIE”. Przełączenie na znaki w negatywie i odwrotnie następuje po naciśnięciu klawisza z symbolem przekreślonego prostokąta (tzw. klawisz Ata-

ri). Przełączanie na pisanie w negatywie (lub odwrotnie) nie jest konieczne dla uzyskania znaków, które są uzyskiwane przy użyciu <ESC>. Są one zawsze wyświetlane w sposób pokazany w tabelach.

Błędy najczęściej występujące przy przepisywaniu programów są powodowane przez wpisanie dużej litery zamiast małej — dotyczy to szczególnie liter P, X i O. Ponadto łatwo pomylić literę O z cyfrą 0, kropkę (.) z przecinkiem (,) oraz średnik (;) z dwukropkiem (:). Podobnie, niektóre znaki specjalne mają wygląd bardzo zbliżony do zwykłych znaków alfanumerycznych. Na wydruku rozróżnić je można tylko według grubości linii. Dla porównania i zapamiętania znaki są pokazane poniżej.

SPECJALNE		NORMALNE	
<	CTRL F	<	znak /
>	CTRL G	>	SHIFT +
=	CTRL N	=	SHIFT -
+	CTRL R	+	znak -
+	CTRL S	+	znak +

NORMALNIE

CTRL ,	CTRL 5
CTRL A	CTRL T
CTRL B	CTRL U
CTRL C	CTRL V
CTRL D	CTRL W
CTRL E	CTRL X
CTRL F	CTRL Y
CTRL G	CTRL Z
CTRL H	ESC ESC
CTRL I	ESC CTRL -
CTRL J	ESC CTRL =
CTRL K	ESC CTRL +
CTRL L	ESC CTRL *
CTRL M	CTRL .
CTRL N	CTRL ;
CTRL O	SHIFT =
CTRL P	ESC CTRL CLEAR
CTRL Q	ESC DELETE
CTRL R	ESC TAB

INVERSE

CTRL ,	CTRL 5
CTRL A	CTRL T
CTRL B	CTRL U
CTRL C	CTRL V
CTRL D	CTRL W
CTRL E	CTRL X
CTRL F	CTRL Y
CTRL G	CTRL Z
CTRL H	ESC SHIFT DELETE
CTRL I	ESC SHIFT INSERT
CTRL J	ESC CTRL TAB
CTRL K	ESC SHIFT TAB
CTRL L	CTRL .
CTRL M	CTRL ;
CTRL N	SHIFT =
CTRL O	ESC CTRL 2
CTRL P	ESC CTRL DELETE
CTRL Q	ESC CTRL INSERT
CTRL R	

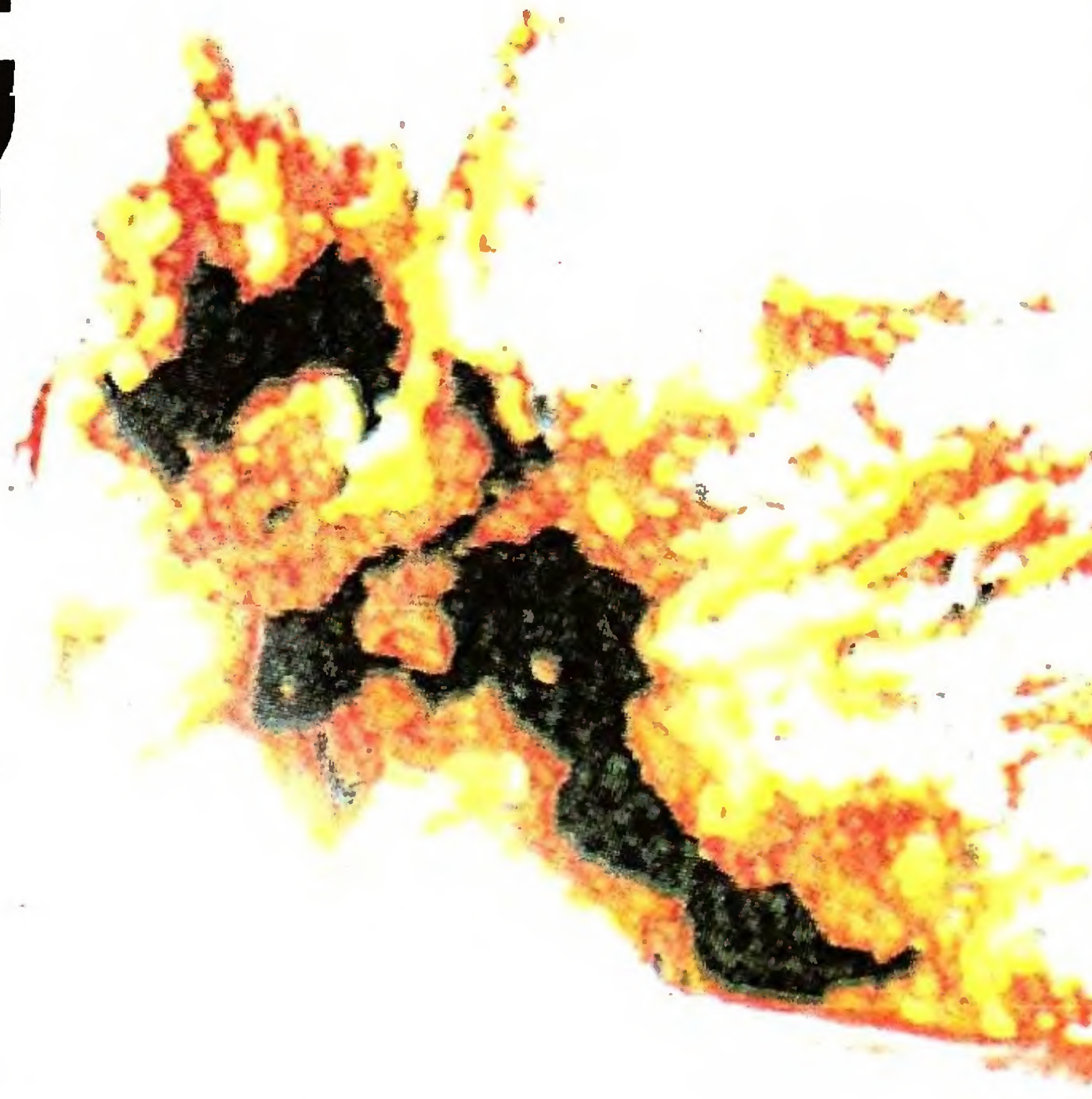
STARCIE

Prezentowana gra ma charakter zręcznościowo-strategiczny. Zabawa polega na niszczeniu wrogów, którzy wdarli się na Twoje terytorium.

Teren podzielono na pola i oznaczono podobnie jak szachownicę współrzędnymi literowo-cyfrowymi. Aby zniszczyć istotę, należy wpisać z klawiatury położenie wroga (w kolejności litera — cyfra). Jeśli źle wpiszesz współrzędne musisz nacisnąć dwa razy klawisz z cyfrą oraz jeden raz klawisz z literą, usłyszysz sygnał dźwiękowy i dopiero teraz możesz wpisywać nowe dane.

Gra kończy się, gdy zabraknie czasu lub bomb (tych ostatnich ubywa po każdym niecelnym strzale, ale jednocześnie po każdym celnym przybywa!).

Jarosław JACEK



SKO

Legenda tej gry podobna jest do legendy gry Robbo. Jednak w odróżnieniu od niej jest to gra platformowa.

Mały stworek został uwięziony w odległym układzie planetarnym. Aby się z niego wydostać, musi przejść szereg planet. Aby wydostać się z planety, musi zebrać śrubki porzucane po całej planecie i zbudować z nich pojazd kosmiczny, którym przeleci na następną planetę.

Naszego bohatera czeka wiele przygód. Bieganie między potworkami i skakanie nad przepaściami to tylko niektóre z atrakcji. Wszędzie czyhają na niego pułapki. Oprócz śrubek znajdzie tam różne przedmioty. Niektóre będą mu przeszkadzały, inne pomagały. Trudząc się z przejściem kolejnej planety musi jednak pamiętać, że każdą planetę można przejść. Oprócz niesamowitego refleksu, do przejścia wszystkich planet potrzebna jest również zdolność logicznego myślenia i umiejętność przewidywania, by nie wskoczyć w pułapkę. Często ważne jest, by nie marnować kluczy, gdy nie jest to potrzebne, bo później mogą się one przydać. Zabawa nie kończy się jednak po przejściu wszystkich planet gdyż tam czekają na niego znajome stworki, które znowu więżą naszego bohatera na planecie, z której zaczynał swą wędrówkę. Biedny Skoczek musi znowu pokonywać cały labirynt z mniejszą tolerancją na pomyłki.

Do gry dołączony jest edytor plansz uruchamiany sześcioliterowym hasłem. Uwaga! Dalsza część artykułu tylko dla znających hasło!

EDYTOR PLANSZ

Aby uruchomić edytor plansz wybieramy na wizytówce opcję „plansze” (klawiszem SELECT). Pojawi się menu planszowe. Wciskamy klawisz „4” (redagowanie) i wpisujemy hasło. Gdy prawidłowo wpisujemy hasło, napisy znikną i po sekundzie pojawi się pierwsza plansza (planeta), lecz zamiast bohatera będzie widoczna strzałka, a pod nią „celownik”. Celownik wskazuje, w którym miejscu na ekranie tekstowym postawimy dany przedmiot. Przedmiot do postawienia wybieramy wskazując na niego strzałką (dolny wiersz). Strzałka stanie się ciemniejsza. Następnie wskazujemy miejsce, gdzie chcemy go postawić i wciskamy przycisk. Wskazując na puste miejsca między przedmiotami uzyskujemy kasowanie (strzałka stanie się jaśniejsza).

Górny wiersz ekranu zawiera tekst „POTWÓR 1 2 3 SKASUJ”. Wskazując strzałką napis „SKASUJ” i wciskając przycisk powodujemy skasowanie całej planszy.

Wskazując strzałką na jedną z liczb w napisie „POTWÓR 1 2 3” spowodujemy miganie potwora o tym numerze. Gdy chcemy zmienić ruch tego potwora wciskamy przycisk. Potwór zniknie, a strzałka pojawi się na środku ekranu. Wciskając po raz drugi przycisk spowodu-

```
DL 9 REM *** J. JACEK ***
CA 1 REM *** STARCIE ***
FO 2 REM (C) 1991 SP. BAJTEK
NI 3 REM
NR 10 GOSUB 5000
PB 20 GRAPHICS 2:SETCOLOR 2,0,0:SETCOLOR
4,0,0:POSITION 2,2: ? H6;"PROSZE POCZEK
AC"REC=0
21 FOR POC=0 TO 400:NEXT POC
DV 25 FOR I=15 TO 0 STEP -0.05:SETCOLOR 0
,0,0:NEXT I
YB 30 PKT=0:EN=100:CZ=10:LO=10
UX 40 DIM D$(3),US(10):D$=""&X":US=""TRAFIO
MYO"
AA 50 GRAPHICS 0:POKE 752,1:POKE 731,1
IF 55 POKE 559,0
EQ 60 SETCOLOR 4,8,1:SETCOLOR 2,8,1:SETCO
LOR 1,0,10
GY 65 POKE 39971,71:POKE 39974,6
SE 70 FOR I=0 TO 13:POKE 39980+I,6:NEXT I
FF 80 POSITION 0,0: ? " & STARCIE"
GK 110 CHBAS=57344
HK 120 CHSET=(PEEK(106)-0)*256
KD 130 FOR I=0 TO 1023
ZZ 140 POKE CHSET+I,PEEK(CHBAS+I)
GB 150 NEXT I
OK 160 POKE 756,CHSET/256
JC 170 READ LZ
JD 180 FOR I=1 TO LZ
KA 190 READ NZ
OS 200 FOR J=0 TO 7
SJ 210 READ A
US 220 POKE CHSET+8*NZ+J,A
GI 230 NEXT J
WX 240 NEXT I:POKE 559,34
OR 250 DATA 5
VO 260 DATA 32,0,0,0,0,0,0,0,0
DK 280 DATA 7,0,127,65,65,65,65,127
BC 290 DATA 6,145,86,56,126,177,108,162,1
45
QJ 300 DATA 5,16,208,57,124,179,168,38,64
XC 310 DATA 3,0,0,0,255,0,0,0,0
UH 400 REM *****
OR 410 REM *****
RY 420 POSITION 1,6: ? "ABCDEFGHIJKLMNORP
5"
XB 430 FOR P=7 TO 12
WB 440 POSITION 0,P: ? P-6;"cccccccccccccc
cccc":NEXT P
HO 450 POSITION 3,3: ? "PUNKTY:0 CZAS:1
0 REKORD:"
FA 460 POSITION 0,5: ? "*****
SZ 470 POSITION 0,13: ? "*****
CR 480 POSITION 2,14: ? "BOMBY:"
OC 490 POSITION 2,15: ? "*****
RP 500 POKE 559,34:NR=1:CLOSE #1:OPEN #1,
4,0,"K:"
XV 510 REM *****
FE 520 FOR POC=0 TO 200:NEXT POC
FI 525 FOR IL=1 TO 3
WG 530 POSITION 18,1: ? " ":FOR POC=0
TO 50:NEXT POC
ZP 540 POSITION 18,1: ? "NR:"NR
KR 550 SOUND 0,200,10,10:POKE 53768,1:FOR
POC=0 TO 50:NEXT POC:SOUND 0,0,0,0
NM 560 NEXT IL:PREDK=150-NR*10:IF PREDK=0
THEN PREDK=10
DA 570 REM *****
XU 580 X=INT(3*NRND(0))+7
YB 590 Y=INT(3*NRND(0))+8
JI 600 POSITION X,Y: ? "&"
XN 610 FOR F=100 TO 110:SOUND 0,F,10,5:50
UND 1,F+1,10,5:NEXT F:SOUND 0,0,0,0:50
UND 1,0,0,0:KLNA=0
MG 620 REM *****
OR 630 FOR POC=0 TO PREDK
ZU 640 B=PEEK(764)
EF 641 IF B<255 AND B<05 THEN GET #1,K
A:KLNA=KLNA+1:SOUND 0,50,10,15:SOUND 0
,0,0,0:GOSUB 700
NJ 645 IF POC/5=INT(POC/5) THEN V=V+1:POS
ITION X,Y: ? D$(V),U:IF V=2 THEN V=0
650 NEXT POC
VH 655 K5=X:Y5=Y
YH 660 K=X+INT(3*NRND(0))-1
SQ 670 Y=Y+INT(3*NRND(0))-1
TE 680 SOUND 0,50,12,12:FOR L=1 TO 3:NEXT
L:SOUND 0,0,0,0
ZI 685 Y=Y+(Y=6)-(Y=13):X=X+(X=0)-(Y=20)
OO 686 CZ=CZ-1:POSITION 20,3: ? CZ:" ":IF
CZ=0 THEN GOTO 2000
ID 687 POSITION X5,Y5: ? "0"
PI 690 GOTO 630
UH 700 IF KLNA=1 THEN X5=KA-64:CLOSE #1:0
UT
```

OCZEK

jemy skasowanie tego potwora z planszy. Kierując joystick w lewo lub w prawo ustalamy, z której strony ma wyjść potwór. Po wciśnięciu przycisku zatwierdzamy ten kierunek. Zamiast strzałki pojawi się wizerunek potwora, którego kierujemy w miejsce nr 1 (wyjaśnienie później). Po wciśnięciu przycisku kierujemy potwora w miejsce nr 2 i wciskamy przycisk. Wtedy potwór wyjdzie zza ekranu ze wskazanej strony (lewej lub prawej) i będzie się kręcił pomiędzy miejscem nr 1 a nr 2. Z miejscami dobrze jest poeksperymentować. Na przykład, aby miejsce nr 1 było w tym samym punkcie co nr 2, albo by miejsce nr 2 było po lewej stronie od miejsca nr 1.

Plansze do redagowania wybieramy wciskając klawisz SELECT.

Pozostaje nam jeszcze opisać.

JAK ZAPAMIĘTAĆ ZMIENIONE PLANSZE

Aby zapamiętać zmienioną planszę wciskamy START. Zamiast strzałki pojawi się bohater, którego kierujemy w miejsce, skąd ma wystartować i wciskamy przycisk. Miejsce startu możemy wybrać dowolne (również za ekranem i wysoko nad platformą) pamiętając jednak o następstwach tego faktu. Następnie na pytanie o liczbę śrubek na planszy wciskamy klawisz z liczbą oznaczającą tę liczbę. Ogranicza to liczbę śrubek na planszy do dziewięciu. Tu także program nie sprawdza, czy podana liczba śrubek zgadza się z ich liczbą na planszy. Oczywiście jest, że gdy podamy liczbę śrubek większą niż ich liczba, nie przejdziemy planszy (chyba, że znajdziemy trochę w pytajnikach). Po wykonaniu podanych czynności plansza zostanie zapamiętana i nastąpi powrót do planszy tytułowej gry. W zapamiętaną planszę możemy już zagrać wybierając jej numer klawiszem HELP i uruchamiając grę.

OPIS PRZEDMIOTÓW

Guma balonowa — wygląda jak pudełko z narysowanym na niej kółkiem. Po wzięciu balonowy Skoczek może latać dopóki się ona nie skończy (biały pasek po lewej stronie).

Gaz — wygląda jak butelka z literką „G”. Po wzięciu gazu Skoczek leci do góry, aż się gaz nie skończy, przy czym nie może się poruszać. Potwory w czasie lotu nie są groźne.

Pytajnik (?) — może dać śrubkę, klucz, gaz, gumę balonową lub „życie”.

Klucz i drzwi — jak nam wszystkim wiadomo drzwi otwieramy kluczem.

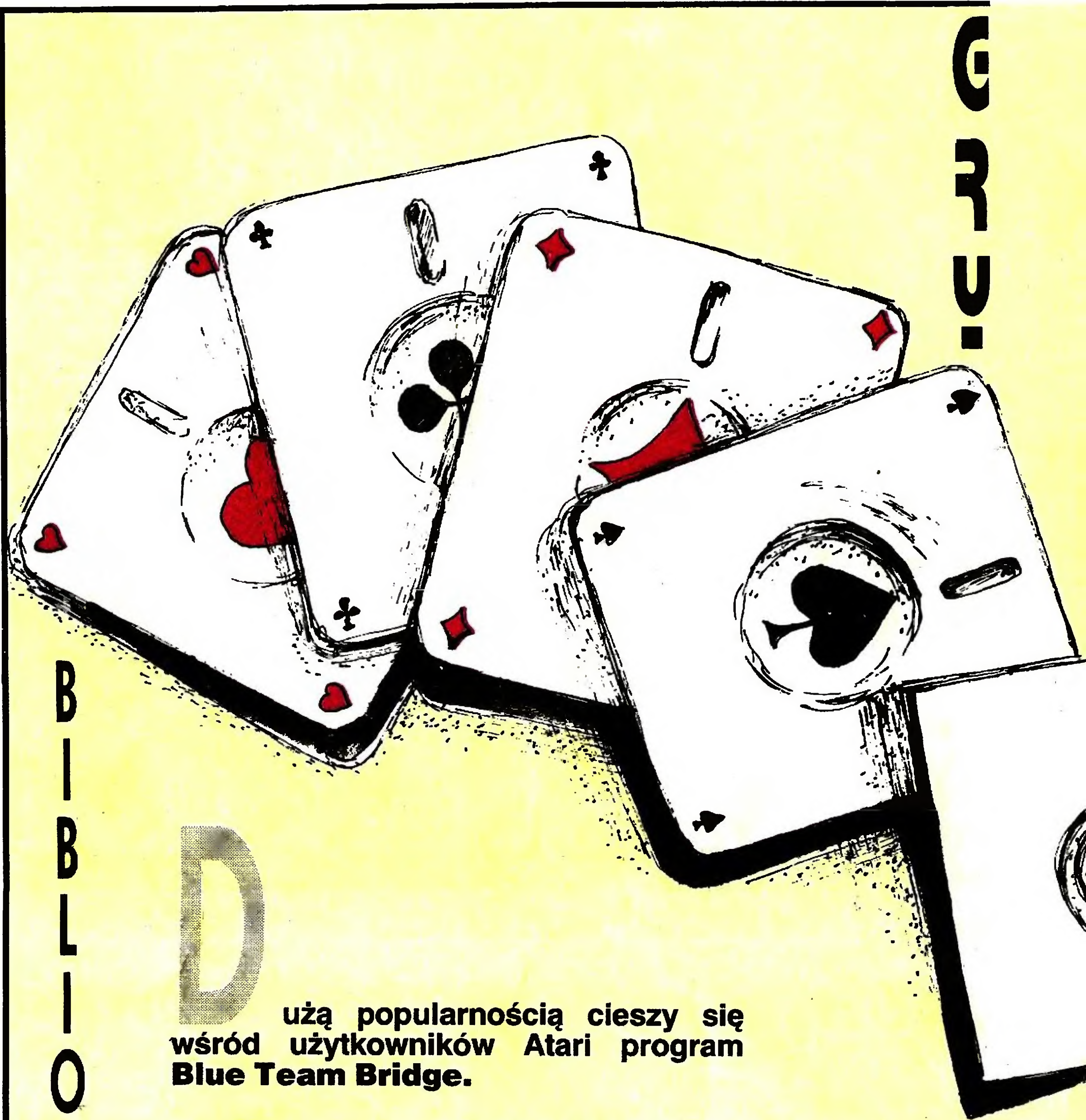
Śrubka — to zbieramy.

Rakieta — do tego wchodzimy.

Życzę miłej zabawy!

Piotr PAWŁOW
(13 lat)

Uwaga: Wielkość tej gry uniemożliwia wydrukowanie jej listingu w „Moim Atari”. Dlatego też publikujemy tylko jej opis, sama gra zaś znajduje się na drugiej stronie dyskietki zawierającej programy z tego numeru naszego pisma (patrz str. 2).



D użę popularnością cieszy się wśród użytkowników Atari program **Blue Team Bridge.**

Korzystając z niego można zapisać na dyskietce maksymalnie do dziesięciu rozdań. Zapisanie następnych wymaga skasowania poprzednich.

Można jednak przechować te rozdzania wykorzystując pokazany tu program (napisany w Turbo Basicu).

Na rozdzania należy przeznaczyć całą stronę dyskietki. Jej przygotowanie wymaga wykonania następujących operacji:

1. Formatujemy dyskietkę przy pomocy DOS 2.5,
2. Uruchamiamy Turbo Basic,
3. Uruchamiamy program BIBLIOTEKA ROZDAŃ,
4. Kasujemy sektory 1, 2, 3, \$168 i (gdy dyskietka jest w rozszerzonej gęstości) \$400.

Ze względu na specyfikę zapisu rozdań, każde z nich zajmuje jeden sektor dyskietki. W zależności od zastosowanej gęstości można więc zapisać na jednej stronie dyskietki 720 (w pojedynczej gęstości) lub 1040 rozdań (w rozszerzonej gęstości).

Możliwości programu BIBLIOTEKA ROZDAŃ są następujące:

1. Zapisanie rozdzania,
2. Przepisanie rozdzania z sektora do pliku, który może być odczytany przez Blue Team Bridge,

3. Zmiana stopnia trudności rozdzania,
4. Kasowanie rozdzania zapisanego w sektorze lub pliku,

5. Odczyt zawartości dyskietki z rozdaniami,
6. Odczyt zawartości plików z rozdaniami,

7. Odszukanie wolnego sektora do zapisu rozdzania.

Na początku każdej części programu ukazuje się napis informujący, która część jest realizowana oraz instrukcja postępowania: «ESC» — powrót, «SPACJA» — kontynuowanie, a niekiedy także «RETURN» — zmiana gęstości dyskietki.

UWAGA: w dalszym opisie przez dyskietkę programu należy rozumieć dyskietkę zawierającą Blue Team Bridge, a przez dyskietkę rozdań — dyskietkę przygotowaną według podanej wyżej instrukcji.

Zapis sektora

1. Podajemy numer rozdzania, pod którym zostało ono zapisane na dyskietce programu.

2. Podajemy w postaci szesnastkowej numer sektora, w którym rozdzanie ma zostać umieszczone na dyskietce rozdań. Od tej chwili rozdzanie to identyfikuje się według

B
I
B
L
I
O
T
E
K
A

R
O
Z
D
A
Ń

BRYDŻOWYCH

numera sektora, w którym jest zapisane.

3. Podajemy stopień trudności rozdania. Proponuję przyjąć następujące zasady:

1 — rozdanie bardzo łatwe, wszystko widoczne jak na dłoni;

2 — rozdanie nieco trudniejsze, wymagana chwila zastanowienia;

3 — rozdanie proste, wymagana umiędzynosić impasowania;

4 — rozdanie średnie, nie zawsze udaje się zrealizować;

5 — rozdanie trudne, sprawa grającemu pewne kłopoty;

6 — rozdanie nieszyblonowe;

7 — rozdanie wzorcowe (uczy nowych sposobów rozgrywki);

8 — szlemiki i szlemy;

9 — kontrakty

nadwyżkowe.

4. Wkładamy dyskietkę programu do stacji i naciskamy **RETURN**. Po chwili ukazują się informacje o rozdaniu.

5. Wkładamy do stacji dyskietkę rozdań i ponownie naciskamy **RETURN**. Rozdanie zostaje zapisane we wskazanym sektorze.

Zapis pliku

Część ta ma działanie odwrotne do opisanego powyżej umożliwia wykorzystanie rozdania zapisanego na dyskietce przez **Blue Team Bridge**.

1. Podajemy numer sektora, z którego ma być odczytane rozdanie.

2. Podajemy numer rozdania, pod którym będzie ono zapisane na dyskietce programu.

Dalsze postępowanie jest analogiczne, jak przy zapisie sektora, lecz odwrotna jest kolejność wkładania dyskietek.

Zmiana stopnia trudności

1. Podajemy numer sektora, w którym chcemy zmienić stopień trudności rozdania.

2. Podajemy nowy stopień trudności.

3. Wkładamy dyskietkę rozdań i naciskamy **RETURN**.

Kasowanie rozdań

Ta część programu umożliwia skasowanie zarówno rozdań zapisanych na dyskietce rozdań, jak i zapisanych na dyskietce programu. W drugim przypadku kasowane są równocześnie wszystkie pliki rozdań.

Odczyt spisu rozdań

W przypadku odczytu spisu z dyskietki rozdań konieczne jest podanie gęstości, w jakiej dyskietka została sformatowana. Można przy tym odczytać spis wszystkich rozdań lub tylko rozdań o wybranym stopniu trudności.

Przy odczycie z dyskietki programu podawane są numery rozdań, które zostały na niej zapisane. Spis jest wyświetlany w takiej postaci, jak stosowana w **Blue Team Bridge**.

Odszukanie wolnego sektora

W tej części program przeszukuje kolejne sektory i wyświetla numer pierwszego wolnego sektora. Dyskietka może być przeszukiwana od początku lub od wskazanego sektora.

Piotr KASTARYNDA

```

YG 0 REM BIBLIOTEKA ROZDAN
QF 1 REM Piotr Kastarynda
DU 2 REM (C) 1991, Sp. Bajtek
NI 3 REM
UM 10 CLR :POKE 65,X3:POKE 82,X0:POKE 502
LF 86,X0:POKE 5028E,X40:POKE 566,158
VS 20 04=X2+X2:05=X2+X3:06=X3+X3:07=X3+Q4
UK 108=X4+Q4:09=Q4+Q5:Q10=Q5+Q5
TL 30 012=Q6+Q6:Q15=Q7+Q8:Q18=Q9+Q9:Q20=Q
CM 10+Q10:Q256=256
JF 40 DIM FILE1$(Q12),FILES$(Q12),TXT$(Q5)
UZ 10 04),A2$(Q4),B5(X3)
JL 50 DIM C$(X1),K$(X2),L$(Q10),M$(X3),SE
UN 100 ? "H" - kasowanie rozdań:Q10 - k
FU 110 ? "Z" - wybór:Q10 - katalog rozdań
IB 120 ? "P" - pierwszy wolny sektor:Q10 - k
MN 130 ? "K" - koniec programu:Q10 - k
XY 140 GET KEY:K=X1:ON KEY-530 GOTO 180,5
OZ 150 GOTO 140
160 CLS :IF SP=X0 THEN POSITION X0,Q10
170 J=Q0:GOTO 200
180 WJ=Q3:GOM WYJSCIE
190 CLS
200 CLOSE :POSITION X0,Q0:?"Numer ro
210 FILES(Q8,Q9)=FILE1$
220 INPUT "H" - kasowanie rozdań:Q10 - k
230 W=X1:GOM SOURCE
240 POKE 5030A,A MOD Q256:POKE 5030B,B
250 OPEN #X1,Q4,X0,FILES:A=PEEK($1404:
260 KODOP=$52:STATOP=$40:EXEC CIO
270 SKOK=X1:I=USR(ADR(TXT$)):POSITION
280 Q10:?"H" - kasowanie rozdań:Q10 - k
290 POKE 5030A,A MOD Q256:POKE 5030B,B
300 KODOP=$50:STATOP=$80:EXEC CIO
310 POKE 5030A,B MOD Q256:POKE 5030B,B
320 IF ZM=X1 THEN WJ=Q6
330 GOM WYJSCIE
340 PROC CIO:POKE 5030A,X1:POKE 5030B,X
350 K=PEEK($A036):K1=K DIV Q10:K2=K MO
360 Q10:PEEK($A037):M=PEEK($A038):IF M
370 IF ZM=X1 THEN SKOK=X1:C1=DEC(C$):C
380 IF (DPEEK($A036)=X0) AND (PEEK($A0
390 IF K2=X0 THEN K$="C"
400 IF K2=X1 THEN K$="D"
410 IF K2=X2 THEN K$="H"
420 IF K2=X3 THEN K$="S"
430 IF K2=Q4 THEN K$="NT"
440 IF L=X0 THEN L$="DBLE"
450 IF L=X1 THEN L$="DBLE"
460 IF L=X2 THEN L$="DBLE"
470 IF (M=X2) OR (M=X0) THEN M$="E-W
480 MS="N-S"
490 IF J=X0 THEN 520
500 POSITION X0,Q10:Q04:?"H" - kasowa
510 ? "Poziom trudności:":C:J=X0
520 ON SKOK GOTO 280,610,880
530 WJ=Q2:GOM WYJSCIE
540 INPUT "K" - kasowanie rozdań:Q10 - k
550 ? "H" - kasowanie rozdań:Q10 - k
560 FILES(Q8,Q9)=FILE1$:FILES(X1,Q7)=""
570 W=X1:GOM SOURCE
580 IF ZM=X1 THEN M=X3
590 KODOP=$52:STATOP=$40:EXEC CIO
600 POKE 5030A,B MOD Q256:POKE 5030B,B
610 LONG=PEEK($A07F):GOM DESTIN
620 CLOSE :OPEN #X1,Q6,X0,FILES:BPUT #
630 WJ=Q4:GOM WYJSCIE
640 ? "H" - kasowanie rozdań:Q10 - k
650 GET KEY:IF KEY=80 THEN SEK$="2D0":
660 IF KEY=82 THEN SEK$="410":?"K":GO
670 GOTO 650
680 5=X1:INPUT "K" - kasowanie rozdań:Q10 - k
690 51=DEC(SEK$):A1=DEC(A1$):B1=A1-X1:
700 POSITION X1,Q18:?"H" - kasowanie rozdań:Q10 - k
710 GET KEY:IF KEY=553 THEN ALL=X1:GOT
720 IF KEY=557 THEN ALL=X0:GOTO 750
730 GOTO 710
740 POSITION Q7,Q20:INPUT "Podaj stopi
750 H=X3:NR=X0:CLS :GOM SOURCE
760 ? "H" - kasowanie rozdań:Q10 - k
770 IF ALL=X1 THEN ? "H":N5T
780 POKE 559,X0:FOR J=X3 TO Q18:KODOP=
790 B1=X1
800 IF (B1>A2) OR (B1>51) THEN POKE 55
810 POKE 5030A,B1 MOD Q256:POKE 5030B,
820 N5=PEEK($A07F)
830 IF (ALL=X1 AND N5T=N5) THEN NR=NR+
840 IF (ALL=X1 AND N5T=N5) THEN EXIT
850 POSITION X0,Q1:IF B1<50100 THEN ? "
860 ? HEX$(B1)
870 SKOK=X3:GOTO 350
880 IF ST=X1 THEN 950
890 POSITION Q4,J:?"H" - kasowanie rozdań:Q10 - k
900 ? "H" - kasowanie rozdań:Q10 - k
910 POSITION Q15,J:?"H" - kasowanie rozdań:Q10 - k
920 IF ALL=X0 THEN POSITION Q18,J:?"H" - kasowanie rozdań:Q10 - k
930 B2=B1+Q10+Q6:B3=(B2 DIV (Q10+Q6))-
940 IF (B1=B4) OR (B1=A2) THEN ST=X0:E
950 ST=X0:NEXT J
960 IF B1=A2 THEN 790
970 IF (ALL=X1 AND NR<>510) THEN 780
980 NR=X0
990 IF (ALL=X1 AND J=Q18) THEN 1000
1000 POKE 559,34:POSITION Q10,Q20+X1
1010 ? "H" - kasowanie rozdań:Q10 - k
1020 GET KEY:IF KEY<>32 THEN 1020
1030 GOTO 760
1040 H WJ=Q5:POKE 5028E,X0:POKE 502B
1050 IF 5=X0 THEN 1070
1060 CLS :IF KEY=535 OR KEY=537 THEN P
1070 CLS
1080 POSITION Q10,Q20:?"H" - kasowanie rozdań:Q10 - k
1090 ? "H" - kasowanie rozdań:Q10 - k
1100 GET KEY
1110 H=X0:IF KEY=27 THEN 60
1120 IF 5=X1 THEN 160
1130 IF KEY=155 THEN ON KEY1-534 GOTO
1140 IF KEY=32 THEN ON WJ GOTO 160,54
1150 GOTO 1100
1160 IF KEY=155 THEN ON KEY1-534 GOTO
1170 IF KEY=32 THEN ON WJ GOTO 160,54
1180 GOTO 1100
1190 WJ=Q5:GOM WYJSCIE
1200 ? "H" - kasowanie rozdań:Q10 - k
1210 GET KEY:IF KEY=80 THEN SEK$="2D0":
1220 IF KEY=82 THEN SEK$="410":?"K":G
1230 GOTO 1210
1240 CLS :POSITION Q9,Q20:?"H" - od po
1250 GET KEY:IF KEY=531 THEN B1=X0:GOT
1260 IF KEY=532 THEN 1280
1270 GOTO 1250
1280 POSITION Q4,Q4:INPUT "Mam szukać
1290 5=X1:51=DEC(SEK$):CLS :M=Q4:GOM S
1300 CLS :POSITION X0,Q10:?"H" - kasowa
1310 ? "H" - kasowanie rozdań:Q10 - k
1320 POSITION Q7,Q12:?"H" - kasowanie rozdań:Q10 - k
1330 IF B1<50100 THEN POSITION Q20+Q10
1340 POSITION Q20+Q10,Q12:?"H" - kasowa
1350 B1=B1+X1:IF B1>51 THEN 1400
1360 POKE 5030A,B1 MOD Q256:POKE 5030B,
1370 IF (DPEEK($A036)<>X0) OR (PEEK($A
1380 POSITION Q5,Q20:?"H" - kasowanie rozdań:Q10 - k
1390 GOTO 1320
1400 POSITION X0,Q10:?"H" - kasowanie rozdań:Q10 - k
1410 GET KEY:IF KEY<>32 THEN 1410
1420 GOTO 60
1430 H SOURCE:TRAP 5SOURCE:POSITION X3
1440 CLOSE :GET KEY:IF KEY<>155 THEN 1
1450 ON W GOTO 240,580,760,1300,1740
1460 H DESTIN:TRAP 5DESTIN:POSITION X3
1470 GET KEY:IF KEY<>155 THEN 1470
1480 ON W GOTO 300,620,300,1590,1770,1
1490 CLS :POSITION Q10+X1,Q10+X1:?"H" - kasowa
1500 GET KEY:IF KEY=84 THEN 1530
1510 IF KEY=78 THEN 60
1520 GOTO 1500
1530 POKE 65,X0:POKE 82,X0:TRAP 40000:
1540 WJ=Q6:ZM=X1:GOM WYJSCIE
1550 INPUT "K" - kasowanie rozdań:Q10 - k
1560 CLS :C1=DEC(C$):GOTO 570
1570 WJ=Q7:GOM WYJSCIE
1580 W=Q4:CLS :GOM DESTIN
1590 CLS :POKE 82,Q12:POSITION Q10+Q4,
1600 IF KEY=82 THEN 1640
1610 OPEN #X1,Q6,X0,FILES:C$=""
1620 FOR I=X0 TO 63:INPUT #X1,FILE1$:F
1630 IF C$="" THEN C1=DEC(C$):C=NEXT I
1640 CLOSE :POKE 82,X0:POSITION Q10,Q2
1650 GET KEY:IF KEY<>32 THEN 1650
1660 GOTO 60
1670 WJ=Q8:GOM WYJSCIE
1680 CLS :POSITION Q12+X1,Q20:?"H" - kasowa
1690 GET KEY:IF KEY=531 THEN 1730
1700 IF KEY=532 THEN CLS :M=Q6:GOM DES
1710 GOTO 1690
1720 DELETE "D:KASOWANIE":WJ=Q8:GOM WJSC
1730 INPUT "K" - kasowanie rozdań:Q10 - k
1740 KODOP=$52:STATOP=$40:EXEC CIO
1750 POKE 5030A,B MOD Q256:POKE 5030B,
1760 DPOKE 5030A,B1 MOD Q256:POKE 5030B,
1770 KODOP=$50:STATOP=$80:EXEC CIO
1780 I=USR(ADR(TXT$)):WJ=Q8:GOM WYJSC
1790 REM ***

```



KASSETOWA BAZA DANYCH

Każdy posiadacz XC12 czy 1010 zna problem braku programów użytkowych pisanych pod kątem magnetofonu.

Praktycznie nie istnieje żadna baza danych, która mogłaby współpracować z tym urządzeniem. Publikowany tu program BasBaz da Ci, czytelniku szansę nieco poważniejszego wykorzystania komputera, chociażby przez skatalogowanie swojej biblioteki.

Oczywiście nie jest to program na miarę SynFile czy HomeCard, ale powinien on być wystarczający do prostych zastosowań. Jeśli uważasz, że to nie wystarcza, musisz niestety pomyśleć o zakupie stacji dysków, bo to już raczej kres możliwości magnetofonu.

Basbaz pozwala na tworzenie zbiorów danych składających się z 700 rekordów. Rekordy są stałej długości (40 znaków) i nie dzielą się na pola. Niedozwolone jest istnienie dwóch takich samych rekordów, co program kontroluje i sygnalizuje.

Basbaz składa się z trzech zasadniczych części. Operacje na rekordach to wpisywanie, poprawianie, sortowanie i kasowanie danych. Opuszczenie poszczególnych wariantów odbywa się przez wciśnięcie «RETURN», bez podania tekstu, chyba że program sygnalizuje coś innego. Każde polecenie należy potwierdzić klawiszem «RETURN». Przy poprawianiu i kasowaniu wystarczy podać fragment tekstu, a program sam odnajdzie i wyświetli stosowny rekord. Niezmienienie rekordu, jeśli wybraliśmy poprawianie, spowoduje sygnalizację błędu, ale bez żadnych następstw. Po wybraniu kasowania, komputer poprosi dodatkowo o potwierdzenie polecenia, przed jego wykonaniem.

Przeglądanie danych jest możliwe jedynie wtedy, gdy baza nie jest pusta, czyli zawiera jakieś rekordy. Przy wyświetlaniu zbioru dowolny klawisz zatrzymuje listę danych, a klawisz «START» przerywa operację. Przy poszukiwaniu wzorcem „d” wyświetlane są wszystkie rekordy zawierające podany ciąg znaków. Na przykład, po

podaniu wzorca „K” wyświetlone zostaną wszystkie rekordy zawierające tę literę. Przy poszukiwaniu wzorcem „s”, wyświetlone zostaną wszystkie rekordy zaczynające się od podanego ciągu. Na przykład, po podaniu wzorca „K” wyświetlone zostaną wszystkie rekordy rozpoczynające się od litery K.

Wariant zapisu i odczytu danych nie wymaga chyba komentarza. Program przechowuje dane w postaci zwykłego pliku ATASCII, zapis odbywa się z krótkimi przerwami. Przy korzystaniu z jakiejś wersji TURBO, trzeba dokonać drobnych poprawek dopasowując instrukcje OPEN do wymaganego przez posiadany system standardu. Gdyby w trakcie zmian zabrakło pamięci, to należy zmienić wielkość zmiennej D\$, w której są przechowywane dane, co oczywiście spowoduje zmniejszenie możliwości do wpisania liczby rekordów.

W programie znajduje się dodatkowa funkcja HELP, która może pomóc w razie wątpliwości.

W celu uzyskania gotowego programu należy korzystając z „Edytora Basica” uważnie przepisać listing 1. Jeżeli przepisanie wierszy 40-80 zawierających podprogramy maszynowe sprawia trudności, to do ich utworzenia można wykorzystać pomocniczy program pokazany na listingu 2. Po przepisaniu (również przy użyciu „Edytora”) radzę najpierw, przed podaniem instrukcji RUN, na wszelki wypadek zapisać go na kasie przez CSAVE. Jeśli wszystko odbyło się poprawnie, to program po wpisaniu instrukcji RUN, na kilkanaście sekund wyłączy ekran, po czym w pamięci znajdzie się jego nowa, przekształcona wersja. Ten nowopowstały program należy zapisać na kasetę instrukcją LIST”C”. Następnie należy wpisać program z listingu 1 pomijając wiersze 40-80 i dołączyć program zapisany przez LIST”C”. Całość zapisujemy na kasetę instrukcją CSAVE i jest to już gotowa końcowa wersja programu BasBaz, którą można uruchomić przy użyciu instrukcji RUN.

Milej pracy!

Jakub CEBULA

LISTING 1

```
EH 0 REM *****
LZ 1 REM * BASBAZ ver. 1.0 *
BL 2 REM * ALIUM VI-XII 1988 *
CY 3 REM * Jakub Cebula *
TA 4 REM * (c) 1990, Sp. Bajtek *
TC 5 REM *
MF 6 REM * 50$ by T.Mrowiec i L.Piela *
AI 7 REM * (c) IKS 1988 *
EP 8 REM *****
JU 10 DIM D$(28000),L$(40),S$(40),50$(142),5Z$(90),DEL$(70),P$(40),LOD$(38),SAV$(27)
IM 20 OPEN #1,4,0,"E":OPEN #2,4,0,"K":P
OKE 566,166
PO 30 S$="" :S$(40)=S$:S$(2)=S$
MX 40 50$(1,76)="hhh, hhh, hhh, khh, lhh, yhh, xh, qh, p, z, o, n, x, u, z, y, k, z, o, t, e, c, u, z, k, u, i, s, k, z, n, p, u, o, t, e, c, p"
JX 50 50$(77,142)="0+1UOT3-P 0+HDLpc0-0-1
H-ZS/LUHET-Uh-TC-Pgh-P f0dpp,%0EOP&%
Z1+P3-
QN 60 SZ$="hhh, kfkhh, oh, nh, mh, lh, oh, p, vilo
ND--ZLi-HLZKi-MEQP%LEPP%ZKI-P -HIL
QNPYDKK%LT-ZM-U-
FG 70 DEL$="hh, hh, lh, oh, n, vllh8%L3, LZXMB-
Hh-VH-ZLi-LZXi-MEOP%LENPR-
CB 80 LOD$="h0-R-D-X-Y-h-u-h-t-B-0-B-X
J-UY-Y-0-
TH 90 SAV$="h0-R-h-Y-h-X-h-u-h-t-B-ALU0-
FL 100 DL=PEEK(560)+256*PEEK(561)
PE 110 GOTO 1490
QT 120 GRAPHICS 0:?:?:? "OPERACJE NA REKO
RDACH"
EC 130 ? :? "Q - WSTAWIANIE"
XH 140 ? :? "W - POPRAWIANIE"
LE 150 ? :? "E - SORTOWANIE"
QH 160 ? :? "R - KASOWANIE"
FO 170 ? :? "T - MENU GLOWNE"
FB 180 ? :? "W bazie ";IR;" rekordow"
TY 190 ? :? "BASBAZ ver. 1.0
*** ALIUM 1988"
YI 200 INPUT #1,L$
TB 210 IF L$="" THEN 270
UW 220 IF L$="E" THEN ? "K":GOTO 370
YJ 230 IF L$="W" THEN 390
RF 240 IF L$="R" THEN 510
PQ 250 IF L$="T" THEN 1490
MP 260 GOTO 120
RN 270 ? "K"
EJ 280 ? "WSTAWIANIE:"
FI 290 POKE 82,0:?:FOR I=0 TO 39:?:?:?:
NEXT I
TI 300 ? "+++":INPUT #1,L$:POKE 82,2
MO 310 IF LEN(D$)>1 THEN KON=320:ER=280:G
OTO 1600
HX 320 IF L$="" THEN 120
QP 330 LN=LEN(L$):IF LN<40 THEN L$(LN+1)=
S$
EQ 340 IR=IR+1:START=(IR-1)*40+1
BN 350 D$(START)=L$:?:?:
PO 360 GOTO 280
IH 370 IF D$="" OR IR=1 THEN 120
SL 380 ? :? "SORTOWANIE ...":X=USR(ADR(50
$),40,0,39,ADR(D$),IR):GOTO 120
KB 390 ? "K":IF LEN(D$)<1 THEN 120
OQ 400 POKE 82,0:?:?:? "PODAJ WZORZEC":IN
PUT #1,L$
LR 410 IF L$="" THEN POKE 82,2:GOTO 120
HW 420 GOSUB 1670
ES 430 IF LEN(L$)=1 AND L$=D$(1,1) THEN R
=1:L$=D$(1,40):GOTO 460
VA 440 IF X=ADR(5Z$) THEN ? :? "NIE MA":
GOTO 400
XJ 450 R=INT((X-ADR(D$))/40)+1:L$=D$(R*40
-39,R*40)
MY 460 ? "REKORD POPRAWIANY:"
OA 470 ? :? L$:?:? "+++":INPUT #1,L$
AT 480 IF L$="" OR L$=" " THEN ? :? "BLA
D - DO USUMANIA SŁUZY INNA OPCJA":GOTO
400
NP 490 ER=400:KON=500:GOTO 1600
XW 500 D$(R*40-39,R*40)=L$:GOTO 400
DL 510 POKE 82,0:?:? "K":IF IR<1 THEN POKE
82,2:GOTO 120
LE 520 GOSUB 630:?:?:? "KASOWANIE REKORDO
W":?:?:? "PODAJ WZOR":INPUT #1,L$:IF L$
="" THEN POKE 82,2:GOTO 120
WZ 530 GOSUB 1670
XE 540 IF LEN(L$)=1 AND L$=D$(1,1) THEN R
=1:GOTO 570
AH 550 IF X=ADR(5Z$) THEN ? :? "NIE MA":
GOTO 520
XM 560 R=INT((X-ADR(D$))/40)+1:L$=D$(R*40
-39,R*40)
ZP 570 ? "REKORD KASOWANY:"
MS 580 ? D$(R*40-39,R*40):?:? "JESTES PE
WNY (T/N)":INPUT #1,L$:IF L$<"T" THEN
GOTO 520
FG 590 X=USR(ADR(DEL$),ADR(D$)+40*R),ADR
(D$)+LEN(D$)+40:D$(LEN(D$)-39)=S$
BE 600 IF LEN(D$)<41 THEN D$="":GOTO 620
CK 610 D$(LEN(D$)-40)=CHR$(PEEK(ADR(D$)+L
EN(D$)-40))
AB 620 ? "KASOWANO":IR=IR-1:GOTO 520
JS 630 IF LEN(D$)<10 THEN POP:POKE 82,2:
GOTO 120
ZK 640 RETURN
AW 650 IF LEN(D$)<1 THEN 1490
RM 660 GRAPHICS 0:?:?:? "PRZEGLADANIE DAN
YCH"
PY 670 ? :? "Q - WYSWIETLENIE ZBIORU"
DC 680 ? :? "W - WYSZUKIWANIE REKORDOW WZ
ORCEM D"
UN 690 ? :? "E - WYSZUKIWANIE REKORDOW WZ
ORCEM S"
```



```

FG 700 ? :? "T - MENU GLOWNE"
ET 710 ? :? "W bazie ";IR;" rekordow"
TQ 720 ? :? "BASBAZ ver. 1.0
    *** ALIUM 1988"

YT 730 INPUT #1,L$
PT 740 IF L$="T" THEN 1490
SN 750 IF L$="Q" THEN 800
AE 760 IF L$="W" THEN 950
EZ 770 IF L$="E" THEN 1080
LZ 780 IF L$="T" THEN 1080
QF 790 GOTO 650
GN 800 GRAPHICS 0:?"WYDRUK ZBIORU:"
OO 810 ? :? "DOWOLNY KLAWISZ - PRZERWA"
XT 820 ? :? "START - MENU"
FP 830 POKE 82,0:?"ZBIOR DANYCH":? :?
    :POKE 764,255
BM 840 A=INT(LEN(D$)/40):K=1
CW 850 KX=K*40:?"D$ (KX-39,KX)
GV 860 IF PEEK(764)<255 THEN POKE 764,25
5:GOSUB 920
WJ 870 IF PEEK(53279)=6 THEN POKE 82,2:GO
TO 650
UC 880 IF K<A THEN K=K+1:GOTO 850
GM 890 ? :? "KONIEC ZBIORU":? :? "NACISN
IJ DOWOLNY KLAWISZ":POKE 764,255
FY 900 IF PEEK(764)=255 THEN 900
PZ 910 POKE 764,255:POKE 82,2:GOTO 650
IG 920 IF PEEK(764)=255 THEN 920
XP 930 POKE 764,255:RETURN
DB 940 ? :? "SZUKANY REKORD":? :? D$(KX-3
9,KX)
IZ 950 GRAPHICS 0:?"WYSZUKIWANIE REKO
RDOW
WEDLUG DOWOLNEGO
WZORCA"
MU 960 POKE 82,0:?"PODAJ WZORZEC":IN
PUT #1,L$:IF L$="" THEN POKE 82,2:GOTO
650
XL 970 GOSUB 1670
KK 980 IF X=ADR(5Z$) THEN ? :? "TAKIE REK
ORDY NIE ISTNIEJA NACISNIJ RETURN"
:INPUT #1,L$:GOTO 960
QU 990 KX=INT((X-ADR(D$))/40)+1)*40:P$=D
$(KX-39,KX)
NT 1000 ? :? "SZUKANE REKORDY:"
GO 1010 ? :? D$(KX-39,KX)
XN 1020 IF (ADR(D$)+LEN(D$))-X<40 THEN 96
0
LS 1030 X=USR(ADR(5Z$),LEN(L$),ADR(L$),AD
R(D$)+KX,ADR(D$)+LEN(D$)+40)
EY 1040 IF X=ADR(5Z$) THEN 960
TL 1050 KX=INT((X-ADR(D$))/40)+1)*40:P$=
D$(KX-39,KX)
RE 1060 ? P$:IF DRUK THEN GOSUB -600
OJ 1070 GOTO 1020
VP 1080 GRAPHICS 0:?"WYSZUKIWANIE REK
ORDOW
WEDLUG STANDARD
WEGO WZORCA"
HE 1090 POKE 82,0:?"PODAJ WZORZEC":I
NPUT #1,L$:IF L$="" THEN POKE 82,2:GOT
O 650
DF 1100 GOSUB 1670
OH 1110 IF X=ADR(5Z$) THEN ? :? "TAKI REK
ORDY NIE ISTNIEJA NACISNIJ RETUR
N":INPUT #1,L$:GOTO 1090
OB 1120 ? :? "SZUKANE REKORDY:"
QC 1130 KX=INT((X-ADR(D$))/40)+1)*40:KX1
=(X-ADR(D$))+40:IF KX1-KX<0 THEN 1150
GS 1140 ? :? D$(KX-39,KX):Q=1
AT 1150 IF (ADR(D$)+LEN(D$))-X<40 AND Q T
HEN Q=0:?"KONIEC":GOTO 1090
SG 1160 IF (ADR(D$)+LEN(D$))-X<40 THEN ?
:?"NIE MA":GOTO 1090
MG 1170 X=USR(ADR(5Z$),LEN(L$),ADR(L$),AD
R(D$)+KX,ADR(D$)+LEN(D$)+40)
JF 1180 IF X=ADR(5Z$) AND Q THEN Q=0:?"
KONIEC":GOTO 1090
MG 1190 IF X=ADR(5Z$) THEN ? :? "NIE MA":
GOTO 1090
OR 1200 GOTO 1130
RV 1210 GRAPHICS 0:?"OPERACJE WEJSCIA
-WYJSCIA"
YV 1220 ? :? "Q - LADOWANIE DANYCH"
YU 1230 ? :? "W - ZAPIS DANYCH"
MQ 1240 ? :? "T - MENU GLOWNE"
VY 1250 ? :? "W bazie ";IR;" rekordow"
JV 1260 ? :? "BASBAZ ver. 1.0
    *** ALIUM 1988"

HS 1270 INPUT #1,L$
OE 1280 IF L$="T" THEN 1490
FT 1290 IF L$="Q" THEN 1320
KL 1300 IF L$="W" THEN 1350
OI 1310 GOTO 1210
XG 1320 GRAPHICS 0:POKE 82,0:?"ODCZYT
- NACISNIJ DOWOLNY KLAWISZ":TRAP 1340
559,0
WD 1325 CLOSE #1:OPEN #1,4,128,"C":POKE
559,0
TU 1330 X=USR(ADR(LOD$),ADR(D$)):IR=INT(X
/40):D$(X+1,X+1)="":POKE 82,2:CLOSE #
1:OPEN #1,12,0,"E":GOTO 1210
WU 1340 GRAPHICS 0:?"WBLAD ODCZYTU":?
"NACISNIJ RETURN"
SR 1345 CLOSE #1:OPEN #1,12,0,"E":INPUT
#1,L$:POKE 82,2:GOTO 1210
WL 1350 IF LEN(D$)<1 THEN 1210
XT 1360 GRAPHICS 0:POKE 82,0:?"
KI 1370 ? :? "ZAPIS - PRZYGOTUJ TASME NACISN
IJ KLAWISZ ":POKE 764,255:CLOSE #1:OP
EN #1,8,128,"C":POKE 559,0
BF 1380 X=USR(ADR(SAV$),LEN(D$),ADR(D$)):
CLOSE #1:OPEN #1,12,0,"E":POKE 82,2:5
OUND 0,0,0,0:SOUND 1,0,0,0:GOTO 1210
RB 1390 GRAPHICS 0:POKE 559,0:POKE 752,1:
POKE 712,4:?">>>HELP"
CR 1400 ? :? "Program ten jest prosta baz
a danych, dopuszczajaca 700 rekordow.
"
ZN 1410 ? :? "W bazie nie ma podzialu na pol
a, nie wolno tworzyć identycznych rek
ordow."
RT 1420 ? :? "Opcje przegladania danych i
kasowania dostepne sa gdy baza nie je
st pusta.:"
FZ 1430 ? :? "Wzorzec s poszukuje bez przesu
niecia, od pierwszej litery rekordu,za
s d wyszukuje dowolny ciag ";
KM 1440 ? :? "znakow.:" :? "Na tasme jest z
apisywana zmienna tekstowa ,rekor
dy sa stalej dlugosci."

```

```

ZK 1450 ? :? "Zyczymy milej pracy !":? :?
"NACISNIJ DOWOLNY KLAWISZ"
EV 1460 POKE 764,255:POKE 559,34
CE 1470 IF PEEK(764)=255 THEN 1470
IU 1480 POKE 732,0:POKE 764,255:GOTO 1490
NK 1490 GRAPHICS 0:POKE 559,0:POKE 710,0:
POKE DL+6,7:POKE DL+7,7:?">>>MENU":P
OKE 752,1
OT 1500 ? :? "START - PRZEGLADANIE DANYC
H"
BO 1510 ? :? "SELECT - OPERACJE NA REKORD
ACH"
YY 1520 ? :? "OPTION - OPERACJE WEJSCIA-W
YJSCIA"
MT 1530 ? :? "HELP - POMOC"
VK 1540 ? :? "ALIUM SKANINA 1988":?"
(C) JAKUB CEBULA":POKE 559,34
OK 1550 IF PEEK(53279)=5 THEN 120
YB 1560 IF PEEK(53279)=6 THEN 650
TK 1570 IF PEEK(53279)=3 THEN 1210
OP 1580 IF PEEK(732)=17 THEN 1390
SU 1590 GOTO 1550
DP 1600 GOSUB 1670
KK 1610 IF X=ADR(5Z$) THEN GOTO KON
TO 1620 KX=INT((X-ADR(D$))/40)+1)*40:P$=
D$(KX-39,KX)
QU 1630 IF LEN(L$)<40 THEN L$(LEN(L$)+1,4
0)=5$
CW 1640 IF L$=P$ THEN ? :? "WBLAD - TAK
I REKORD JUZ ISTNIEJE":? :GOTO ER
DD 1650 X=USR(ADR(5Z$),LEN(L$),ADR(L$),KX
+40,ADR(D$)+LEN(D$))
QZ 1660 GOTO 1610
JE 1670 X=USR(ADR(5Z$),LEN(L$),ADR(L$),AD
R(D$),ADR(D$)+LEN(D$)+40):RETURN

```

LISTING 2

```

QF 10 REM BASBAZ - PROGRAM POMOCNICZY
WQ 20 REM Jakub Cebula
SS 30 REM (c) 1990, Sp. Bajtek
BB 40 REM
CL 50 GRAPHICS 0:POKE 559,0:POSITION 2,3
SX 60 ? "40 SO$(1,76)=";CHR$(34);:FOR I=1
TO 76:READ A:?"CHR$(27);CHR$(A);:NEXT
I:?"CHR$(34)
VG 70 ? "50 SO$(77,142)=";CHR$(34);:FOR I
=77 TO 142:READ A:?"CHR$(27);CHR$(A);:
NEXT I:?"CHR$(34)
UR 80 ? "60 SZ$=";CHR$(34);:FOR I=1 TO 85
:READ A:?"CHR$(27);CHR$(A);:NEXT I:?"C
HR$(34)
JB 90 ? "70 DEL$=";CHR$(34);:FOR I=1 TO 6
0:READ A:?"CHR$(27);CHR$(A);:NEXT I:?"
CHR$(34)
XU 100 ? "80 LOD$=";CHR$(34);:FOR I=1 TO
38:READ A:?"CHR$(27);CHR$(A);:NEXT I:?"
CHR$(34)
AQ 110 ? "90 SAV$=";CHR$(34);:FOR I=1 TO
27:READ A:?"CHR$(27);CHR$(A);:NEXT I:?"
CHR$(34)
CX 120 ? "POKE 842,12:GR.0":POSITION 0,0:
POKE 842,13:NEW
VL 130 REM SORTOWANIE T.Piela i L.Mrowiec
EG 140 DATA 104,104,104,133,205,104,104,1
33,206,104,104,133,203,104,104,133,204
,104,133,217,104,133,216,104,133
VY 150 DATA 209,104,133,208,169,0,133,218
,133,207,162,1,165,216,133,214,165,217
,133,215,24,165,214,133,212,101,205
LQ 160 DATA 133,214,165,215,133,213,105,0
,133,215,164,203,165,206,240,10,177,21
4,209,212,144,44,240,12,176,19
FI 170 DATA 177,214,209,212,144,13,240,2
,176,30,200,196,204,240,227,176,23,144
,223,169,1,133,218,164,205,136,177
MY 180 DATA 214,72,177,212,145,214,104,14
5,212,192,0,208,241,232,224,0,208,2,23
0,207,228,208,208,172,165,209
LV 190 DATA 197,207,208,166,165,218,201,0
,208,144,96
WX 200 REM SZUKANIE i dalsze J.Cebula
QG 210 DATA 104,104,104,133,203,198,203,1
04,133,207,104,133,206,104,133,205,104
,133,204,104,133,209,104,133,208
YL 220 DATA 160,0,177,204,209,206,240,24
,24,165,204,105,1,133,204,165,205,105,0
,133,205,197,209,208,231
YR 230 DATA 165,204,197,208,208,225,96,16
5,203,201,0,240,13,160,0,200,177,204,2
09,206
YU 240 DATA 208,217,196,203,208,245,165,2
04,133,212,165,205,133,213,96
YV 250 REM KASOWANIE
XD 260 DATA 104,104,133,205,104,133,204,1
04,133,207,104,133,206,160,0,177,204,7
2,56,165,204,233
SX 270 DATA 40,133,204,165,205,233,0,133
,205,104,160,0,145,204,24,165,204,105,4
1,133,204,165,205,105
KD 280 DATA 0,133,205,197,207,208,216,165
,204,197,206,208,210,96
UO 290 REM BLOAD
OY 300 DATA 104,169,7,141,82,3,169,255,14
1,88,3,141,89,3,104,141,85,3,104,141
VG 310 DATA 84,3,162,16,32,86,228,173,88
,3,133,212,173,89,3,133,213,96
YX 320 REM BPUT
PX 330 DATA 104,169,11,141,82,3,104,141,8
9,3,104,141,88,3,104,141,85,3,104,141
,84,3,162,16,76,86,228

```

RUCH STATKU

SZKÓŁKA

Do tej pory projektowaliśmy głównie operacje stanowiące przygotowanie do właściwej rozgrywki. Dopiero teraz możemy zająć się zasadniczą częścią naszego programu.

Przypominam, że gra właściwa jest pętlą, w której kolejno są wywoływane następujące procedury:

- Ruch Statku
- Ruch Niszczyciela
- Ruch Okrętu Podwodnego
- Odpalenie Torpedy
- Peryskop
- Ruch Okrętu Podwodnego
- Peryskop

Procedury te są znacznie bardziej skomplikowane od dotychczas omawianych, więc będziemy musieli każdą z nich przed zakodowaniem rozwinąć w schemat blokowy. Taki tok postępowania ułatwi zapisanie programu w formie zrozumiałej dla komputera.

Pierwszym elementem tego cyklu jest procedura realizująca przemieszczanie się statku. Należy więc obliczyć kierunek ruchu tak, aby statek nie opuścił planszy i odpowiednio zmodyfikować zmienne opisujące jego położenie. Konieczne jest przy tym sprawdzenie, czy statek nie znajdzie się na polu zajętych już przez jeden z niszczycieli. W takim przypadku faza obliczania parametrów ruchu statku musi być powtórzona.

Po wykonaniu przez statek ruchu może on znaleźć się nad okrętem podwodnym. Sytuacja taka z pewnością będzie rozpoznana przez okręt podwodny, więc odpowiednia informacja musi zostać wyświetlona na ekranie. Drugi — ważniejszy — przypadek to znalezienie się statku w porcie. Powoduje to automatyczne zakończenie gry. Sygnalizujemy to nadając ustaloną wartość (tu 4) zmiennej „q” i opuszczając procedurę.

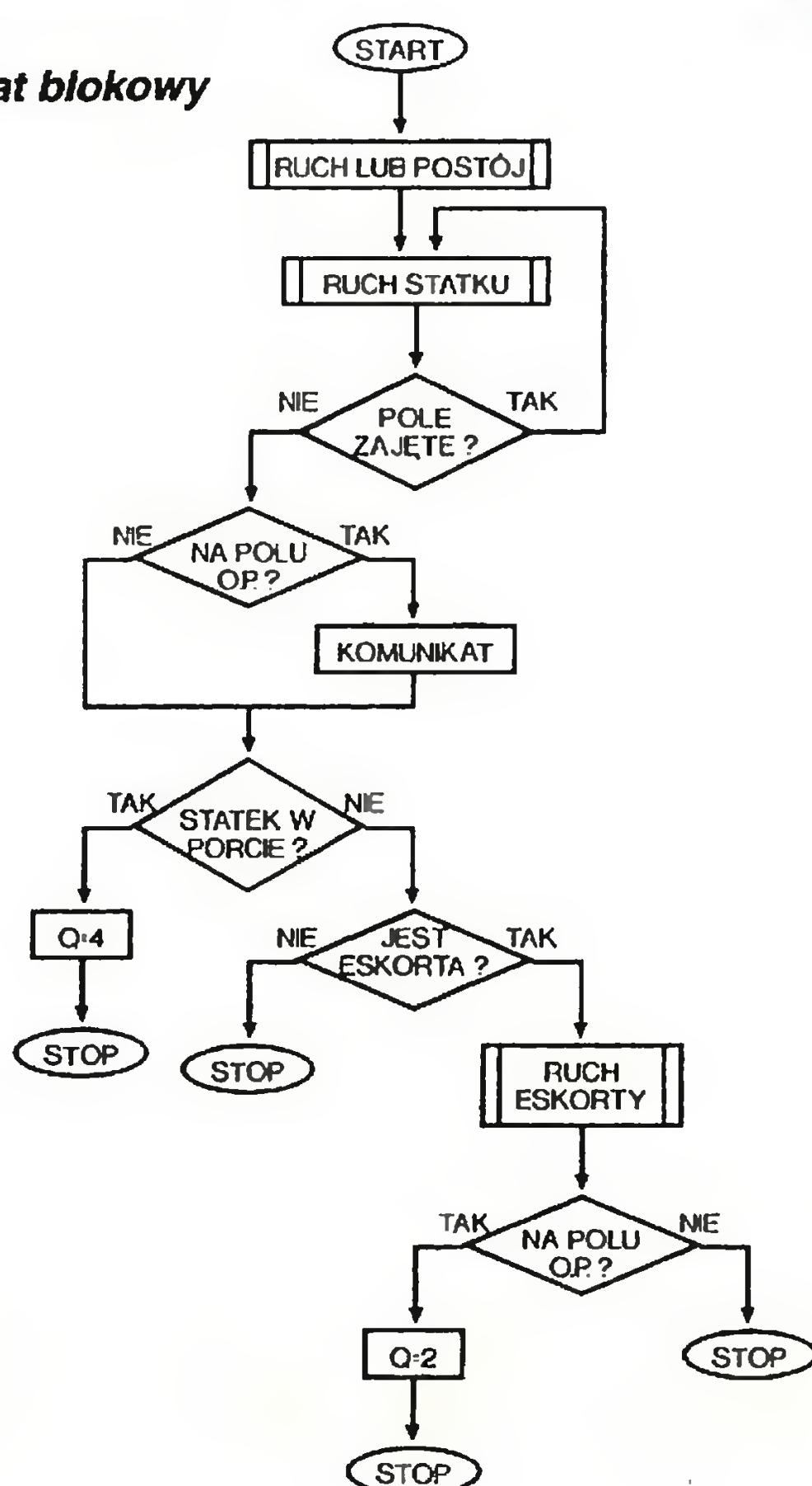
Następnie należy sprawdzić, czy nie został uprzednio zatopiony niszczyciel eskortujący. Jeżeli jeszcze istnieje, to ustalamy jego nowe położenie w pobliżu statku (na polu sąsiadującym). Teraz również trzeba sprawdzić, czy na tym samym polu nie znajduje się okręt podwodny. Jeżeli tak, to zakładamy, że niszczyciel wykrywa szykujący się do ataku okręt i niszczy go bombami głębinowymi. Sygnalizujemy tą ewentualność nadając zmiennej „q” wartość 2 i kończąc procedurę.

Ponieważ statek dążąc do portu mógłby zbyt szybko umknąć nie dając szans na zniszczenie, należy wprowadzić dodatkowe ograniczenie. Będzie ono polegać na tym, że statek przybywając na redę portu może oczekiwać na miejsce i w związku z tym przez kilka (1—3) faz nie wykonuje ruchu. Po uwzględnieniu tego ograniczenia schemat blokowy naszej procedury będzie wyglądał jak na rys. 1.

Rozwijając ten schemat należy przede wszystkim pamiętać o wprowadzeniu warunków sprawdzających, czy któraś z jednostek nie znalazła się poza planszą gry. Jeśli taki przypadek zachodzi można zmienić kierunek ruchu na przeciwny (tak jest dla statku) lub powtórzyć całe obliczanie nowej pozycji (tak jest dla niszczyciela eskorty). Różne podejście do tego zagadnienia wynika z faktu, że statek porusza się po planszy swobodnie, niszczyciel zaś jest związany ze statkiem i zawsze musi się znajdować w jego pobliżu.

Uważny czytelnik analizując wydruki procedur w **Action!** (listing 1) lub w **Basicu** (listing 2) zauważy z pewnością, że ruch statku jest poddany dodatkowemu ograniczeniu. Aby statek nie dążył do portu najkrótszą możliwą drogą, zostało bowiem dodane losowe (z prawdopodobieństwem 25%) zmienianie kierunku ruchu w pionie i w poziomie.

Rys. 1
Schemat blokowy



Szybkość pracy komputera znacznie przekracza możliwości percepcji człowieka. Aby możliwe było odczytanie wyświetlanych przez program komunikatów, należy wprowadzić procedurę wstrzymującą na pewien okres czasu wykonywanie programu. Procedura ta jest w **Action!** następująca:

```

PROC Opoznienie(CARD I)
CARD J
FOR J=1 TO I DO OD
RETURN
  
```

Została ona zrealizowana w najprostszy możliwy sposób, czyli przez zastosowanie pętli. Dla umożliwienia regulacji opóźnienia granicę pętli podaje się jako parametr procedury. Nieco inaczej wygląda to w **Basicu**, gdyż nie można tam przekazać parametru:

```

RK 108 REM
QH 109 REM *** Opoznienie ***
GS 110 FOR L=1 TO DL:NEXT L:RETURN
  
```

W tym przypadku długość opóźnienia musi być ustalona przed wywołaniem procedury przez nadanie odpowiedniej wartości zmiennej „DL”.

Porównując wywołania procedury opóźniającej w obu stosowanych językach łatwo można zauważyć znaczną różnicę wartości parametru. Wynika ona z różnej szybkości działania **Basica** i **Action!**. Nie oznacza to oczywiście, że program w **Action!** jest ponad 100 razy szybszy. Jednak w przypadku pętli taka różnica występuje. Pojawia się jeszcze pytanie, jak określić wartości opóźnienia. Można próbować je wyliczyć, ale najprostszym sposobem jest doświadczalne ustalenie podczas próbnego uruchamiania programu.

LISTING 1

```

PROC RuchStatku()
BYTE s,n,k,nx,ny
IF sx<2 AND sy<2 THEN
s=-1
ELSE
n=Rand(100)
IF n>40 THEN
s=3
ELSEIF n>20 THEN
s=2
ELSEIF n>5 THEN
s=1
FI
Czyszczenie()
PrintE("Statek wykonał ruch.")
Opoznienie(30000)
FI
IF s=0 THEN
RETURN
FI
DO
nx=sx ny=sy
IF Rand(100)<75 AND nx>0 OR nx>8 THEN
nx=-1
ELSE
nx=+1
FI
IF Rand(100)<75 AND ny>0 OR ny>8 THEN
ny=-1
ELSE
ny=+1
FI
IF (nx#dx(1) OR ny#dy(1)) AND
(nx#dx(2) OR ny#dy(2)) THEN
EXIT
FI
OD
ox=sx oy=sy sx=nx sy=ny
Kasowanie(ox,oy)
IF sx=mx AND sy=my THEN
PrintE("Statek jest nad okrętem podwodnym.")
Opoznienie(30000)
FI
IF sx=0 AND sy=0 THEN
q=4 RETURN
FI
IF ns(2)#0 THEN
Kasowanie(dx(2),dy(2))
DO
dx(2)=sx+1-2*Rand(2)
UNTIL dx(2)<10 OD
DO
dy(2)=sy+1-2*Rand(2)
UNTIL dy(2)<10 OD
IF dx(2)=mx AND dy(2)=my THEN
Opoznienie(10000)
q=2
FI
FI
RETURN
  
```

LISTING 2

```

SC 497 REM
JL 498 REM *** Ruch Statku ***
SI 499 REM
WW 500 IF SX<2 AND SY<2 THEN S=S-1:GOTO 520
OT 510 N=RND(0):S=(N>0.4)+(N>0.2)+(N>0.05)
KM 520 GOSUB 40:? "Statek wykonał ruch.":
DL=200:GOSUB 110
UX 530 IF S=0 THEN RETURN
XO 540 NX=SX:NY=SY:CX=1:CY=1
HD 550 IF RND(0)<0.75 AND NX>0 OR NX>8 THEN
CX=-1
QY 560 NX=NX+CX:IF RND(0)<0.75 AND NY>0 OR
NY>8 THEN CY=-1
HI 570 NY=NY+CY:IF NX=DX(0) AND NY=DY(0)
OR NX=DX(1) AND NY=DY(1) THEN 540
OE 580 OX=SX:OY=SY:SX=NX:SY=NY:GOSUB 50
LK 590 IF SX=MX AND SY=MY THEN ? "Statek
jest nad okrętem podwodnym.":DL=200:GO
SUB 110
JH 600 IF SX=0 AND SY=0 THEN Q=4:RETURN
QC 610 IF NS(1)=0 THEN RETURN
QK 620 OX=DX(1):OY=DY(1):GOSUB 50
RM 630 DX(1)=SX+1-2*INT(RND(0)*2):ON DX(1)
<0 OR DX(1)>9 GOTO 630
XT 640 DY(1)=SY+1-2*INT(RND(0)*2):ON DY(1)
<0 OR DY(1)>9 GOTO 640
BN 650 IF DX(1)=MX AND DY(1)=MY THEN DL=1
OO:GOSUB 110:Q=2
ZO 660 RETURN
  
```

Wojciech ZIENTARA

Na początek proponuję przepisać kolejno programy 1 i 2 oraz nagrać je na kasiecie. Następnie wczytujemy program 1. Po podaniu adresu kodowania (zwracam uwagę na właściwy wybór miejsca w pamięci — musi mieć ono stałą zawartość na przestrzeni co najmniej 256 bajtów, bo inaczej nie uda nam się odkodować pliku) zostanie zapisanych na kasiecie kilka rekordów. Kolejną czynnością jest wczytanie programu 2, w którym podajemy ten sam adres co uprzednio, piszemy NEW i teraz wczytujemy zabezpieczany program. Wykonując X=USR(1536) nagrywamy zaraz za zapisanymi wcześniej rekordami (nie zostawiając żadnej przerwy) dwa kolejne bloki pliku. Całość wczytujemy włączając komputer z wciśniętym klawiszem <START>. W czasie tej operacji na ekranie mogą wystąpić pewne efekty, ale nie należy się nimi przejmować.

W świadomości większości użytkowników Atari pokutuje pogląd, że w sprawie ochrony własnych dzieł w Basicu niewiele można zrobić. Rzeczywiście, przeciętny użytkownik korzystający jedynie z wbudowanego Basica nie może liczyć na sukcesy. Język ten utrudnia lub nawet uniemożliwia to zadanie. Jedynym sposobem jest posłużenie się asemblerem przy odpowiednim wykorzystaniu możliwości komputera.

Jeżeli mimo to nie jesteśmy pewni zabezpieczenia, to przed wykonaniem USR należy wpisać wiersz 10 z programu 3, ale bez numeru wiersza. Spowoduje on, że listing zabezpieczanego programu stanie się całkowicie nieczytelny. Jeżeli mamy zamiar dokonywać jeszcze jakichś zmian w programie, to przed użyciem programu 3 trzeba zapisać źródłową wersję programu w Basicu. Zmiana postaci listingu jest w zasadzie nieodwracalna.

Prezentowane w tym odcinku zabezpieczenie utrudnia dostanie się do samego programu, jak i jego kopiowanie. Wykorzystuje ono właściwości opisane w poprzednim numerze, szczegółów zdradzać nie będę, bo szybko okazałoby się ono nieskuteczne. Sądzę, że nikt z amatorów swego nazwiska w cudzych programach połamie sobie na nim zęby.

Za dwa miesiące przedstawię ten sam temat, lecz dla posiadaczy stacji dysków.

Andrzej ZALEWSKI

UWAGA: Przed utworzeniem wersji zabezpieczonej należy zapisać wersję źródłową. Użycie programów zabezpieczających usuwa z pamięci program w Basicu.

XN	0 REM TYLKO DLA ZAAWANSOWANYCH (4)	5,146,63,181,183,24,180,180,24,249,115
KT	1 REM PROGRAM 1	,233,76,2607
LU	2 REM Andrzej Zalewski	JF 390 DATA 191,30,63,76,194,193,106,68,7
DV	3 REM (c) 1991, Sp. Bajtek	2,205,196,132,144,46,203,204,205,206,2
MB	10 GRAPHICS 0: ? "Proszę przygotować ka setę i czekać."	07,208,209,3158
UP	20 RESTORE 200:Q=21000:W=21100:L=190:G OSUB 1000	UD 400 DATA 210,211,212,213,214,215,216,2
JJ	30 RESTORE 300:Q=1536:W=1770:L=290:GOS UB 1000	17,218,219,220,221,222,223,224,225,226
YS	40 RESTORE 500:Q=16384:W=16510:L=490:G OSUB 1000	,227,228,229,230,4620
FU	50 RESTORE 600:Q=20000:W=20105:L=590:G OSUB 1000	TQ 410 DATA 231,232,233,234,235,236,237,2
JA	60 ? "Podaj adres kodowania ";:INPUT N :N1=INT(N/256):N0=N-N1*256:POKE 112,N0 :POKE 113,N1	38,239,240,241,242,243,244,245,246,247
CT	80 ? "Naciśnij PLAY/REC i dowolny klaw isz"	,248,249,250,251,5061
LA	100 OPEN #1,8,128,"C":X=USR(16384):PO KE 649,0:CLOSE #1:END	WY 420 REM END
WZ	190 REM QUIT	VC 490 REM ZAPIS
EC	200 DATA 169,159,143,53,6,172,146,138, 205,11,163,8,174,13,174,61,157,59,16,5 1,72,2150	XL 499 REM PROGRAM ZAPISU + CODE
AG	210 DATA 241,187,61,26,201,225,185,28, 147,49,29,166,115,159,144,191,184,46,1 86,192,201,2963	VQ 500 DATA 104,162,0,189,32,78,157,0,4,2
NN	220 DATA 52,251,217,132,35,162,122,50, 82,20,6,59,38,45,8,22,17,30,36,41,44,1 469	32,208,247,169,252,32,124,254,169,252, 32,124,2821
IH	230 DATA 51,81,83,88,113,113,107,70,71 ,72,73,74,75,76,77,78,79,80,81,82,83,1 707	CA 510 DATA 254,165,112,73,122,141,122,6, 165,113,73,123,141,123,6,169,85,141,25
FS	240 DATA 84,85,86,87,88,89,90,91,92,93 ,94,95,96,97,98,99,100,101,102,103,104 ,1974	4,5,141,2534
BG	290 REM LOADBASIC	TW 520 DATA 255,5,169,254,141,4,3,169,5,1
CN	299 REM LOADBASIC	41,5,3,169,87,162,255,160,0,32,95,64,2 178
DS	300 DATA 104,168,150,142,193,7,175,153 ,133,63,8,162,56,128,12,220,165,18,159 ,57,22,2315	OK 530 DATA 162,0,189,8,82,157,0,4,232,20
GR	310 DATA 183,22,183,96,57,70,255,177,5 5,28,207,219,136,223,174,96,39,171,35, 43,128,2597	8,247,169,252,32,124,254,96,104,160,0, 132,2612
PO	320 DATA 41,166,41,46,140,19,144,49,18 2,97,157,103,22,170,82,155,58,134,44,5 7,171,2058	HH 540 DATA 116,177,112,69,116,145,112,20
H1	330 DATA 191,168,161,106,147,178,232,7 0,67,205,57,199,79,79,224,79,75,213,32 ,223,86,2871	0,208,245,96,141,2,3,142,8,3,140,9,3,1 69,2216
LY	340 DATA 87,251,84,83,244,90,94,242,14 ,125,185,89,192,97,207,103,96,244,22,2 02,109,2870	RF 550 DATA 96,141,0,3,169,0,141,1,3,169, 35,141,6,3,173,2,3,160,64,201,82,1593
RW	350 DATA 109,162,250,28,200,238,234,0, 212,243,246,5,213,118,198,8,32,122,133 ,237,13,3001	QM 560 DATA 240,2,160,128,140,3,3,169,128
PN	360 DATA 182,175,118,103,243,38,245,64 ,23,87,100,32,139,14,133,36,142,2,212, 147,59,2294	,141,11,3,76,89,228,0,0,0,0,0,0,1521
EP	370 DATA 19,17,153,63,12,29,148,186,23 9,120,16,156,156,46,169,161,47,173,166 ,15,199,2290	PH 570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0
IU	380 DATA 37,169,169,2,172,32,175,172,2	VJ 590 REM BOOT
		BD 600 DATA 0,1,0,96,64,96,169,254,141,4, 3,169,5,141,5,3,169,82,162,255,160,197 9
		CQ 610 DATA 0,32,79,96,169,0,133,116,169, 6,133,117,32,65,96,32,141,253,169,0,13 3,1971
		BD 620 DATA 116,169,4,133,117,32,65,96,16 2,0,189,0,4,157,128,155,232,208,247,76 ,0,2290
		YC 630 DATA 6,96,160,0,177,116,132,112,69 ,112,145,116,200,208,245,96,141,2,3,14 2,8,2286
		LA 640 DATA 3,140,9,3,169,96,141,0,3,169, 0,141,1,3,169,35,141,6,3,173,2,1407
		XM 650 DATA 3,160,64,201,82,240,2,160,128 ,140,3,3,169,128,141,11,3,76,89,228,0, 2031
		PG 660 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0 ,0,0,0,0,0,0,0
		NK 1000 FOR N=Q TO W STEP 21
		EZ 1010 L=L+10:S=0
		EQ 1020 FOR M=0 TO 20
		ON 1030 READ A:S=S+A:POKE N+M,A
		PS 1040 NEXT M:READ SS:IF SS<>S THEN ? "B LAD W WIERSZU ";L:END
		HH 1050 NEXT N
		AR 1060 RETURN

```

XM      0 REM TYLKO DLA ZAAWANSOWANYCH (4)
LI      1 REM PROGRAM 2
LU      2 REM Andrzej Zalewski
DV      3 REM (c) 1991, Sp. Bajtek
XC      10 GRAPHICS 0; "Proszę czekać!"
WB      15 L=190:FOR N=1536 TO 1777 STEP 21
VD      20 L=L+10:S=0
HO      30 FOR M=0 TO 20
PN      40 READ A:S=S+A:POKE N+M,A
NY      50 NEXT M:READ SS:IF SS<>S THEN ? "BLA
D W WIERSZU ";L:END
KN      60 NEXT N
ES      70 POKE 1788,89:POKE 1789,228
ZH      80 ? "Podaj adres kodowania ";:INPUT N
N1=INT(N/256):NO=N-N1*256
WW      90 POKE 1552,NO:POKE 1553,N1:END
CZ      200 DATA 104,160,0,165,128,133,112,165
,129,133,113,160,0,177,112,89,0,254,14
5,112,200,2591
OW      210 DATA 208,246,230,113,165,113,197,1
45,208,236,169,52,141,2,211,169,204,14
1,4,210,169,3333
CN      220 DATA 5,141,6,210,169,96,141,42,2,1
41,0,3,32,104,228,169,3,162,0,160,250,
2064
UX      230 DATA 32,92,228,173,42,2,208,251,16
9,253,141,4,3,169,3,141,5,3,162,0,181,
2262
TN      240 DATA 128,157,16,4,232,224,50,208,2
46,165,128,56,233,2,133,112,141,0,4,16
5,129,2533
WG      250 DATA 233,0,133,113,141,1,4,165,144
,56,229,112,141,2,4,165,145,229,113,14
1,3,2274
MH      260 DATA 4,160,0,177,112,141,4,4,200,1
77,112,141,5,4,162,60,160,0,169,85,141
,2018
ZR      270 DATA 253,3,141,254,3,169,87,32,208
,6,173,0,4,141,4,3,173,1,4,141,5,1805
AY      280 DATA 3,174,2,4,169,96,141,0,3,169,
85,160,0,145,112,200,145,112,172,3,4,1
899
EQ      290 DATA 169,57,32,208,6,169,60,141,2,
211,169,0,141,4,210,141,6,210,96,141,2
,2175
ZA      300 DATA 3,142,8,3,140,9,3,169,96,141,
0,3,169,0,141,1,3,169,35,141,6,1382
SY      310 DATA 3,173,2,3,160,64,201,82,240,2
,160,128,140,3,3,169,128,141,11,3,76,1
892

```

```
XH 0 REM TYLKO DLA ZAAWANSOWANYCH (4)
LX 1 REM PROGRAM 3
LU 2 REM Andrzej Zalewski
DV 3 REM (c) 1991, S. Bajtek
PH 10 ST=PEEK(130)+256*PEEK(131):EN=PEEK(
132)+256*PEEK(133):FOR N=ST TO EN:POKE
N,155:NEXT N
```

```
(* Biblioteka funkcji dodatkowych *)
(* Jakub Cebula, Przemysław Osocha *)
(* (c) 1990, Sp. Bajtek *)

(* Kod maszynowy *)
#A
;
;   Definicje funkcji VHstick
;
STICKDEF LDY #9
LDA (_SP),Y
TAX
LDA #D300
P CPX #0
BEQ EX
LSR A
LSR A
DEX
JMP P
EX AND #3
LDY #9
STA (_SP),Y
RTS
#

(* Blok definicji PASCALA *)

procedure poke(adres:real;co:integer);
var
  gdzie:integer;
begin
  if adres>32767 then
    gdzie:=trunc(adres-65536.0)
  else
    gdzie:=trunc(adres);
#A
LDY #5
LDA (_SP),Y
STA _T
INY
LDA (_SP),Y
STA _T+1
LDY #7
LDA (_SP),Y
LDY #0
STA (_T),Y
#
end;

procedure dpoke(adres,co:real);
var
  x:integer;
  y:real;
begin
  x:=trunc(co/256);
  poke(adres+1,x);
  y:=x*256.0;
  x:=trunc(co-y);
  poke(adres,x);
end;

function peek(gdzie:real):integer;
var
  x:integer;
begin
  if gdzie>32767 then
    x:=trunc(gdzie-65536.0)
  else
    x:=trunc(gdzie);
#A
LDY #5
LDA (_SP),Y
STA _T
INY
LDA (_SP),Y
STA _T+1
LDY #0
LDA (_T),Y
LDY #5
STA (_SP),Y
INY
LDA #0
STA (_SP),Y
#
  peek:=x;
end;

function dpeek(gdzie:real):real;
begin
  dpeek:=peek(gdzie)+256.0*peek(gdzie+1);
end;

function Vstick(joy:integer):integer;
begin
  if joy<>0 then joy:=2;
#A
JSR STICKDEF
#
  case joy of
    1: Vstick:=-1;
    2: Vstick:=1;
    3: Vstick:=0;
  end;
end;

function Hstick(joy:integer):integer;
begin
  if joy=0 then joy:=1
  else joy:=3;
#A
JSR STICKDEF
#
  case joy of
    1: Hstick:=-1;
    2: Hstick:=1;
    3: Hstick:=0;
  end;
end;

function stick(joy:integer):integer;
begin
  if joy=0 then stick:=peek(632.0)
  else stick:=peek(633.0);
end;

function strig(joy:integer):integer;
begin
  if joy=0 then strig:=peek(644.0)
  else strig:=peek(645.0);
end;
```

PEEK, POKE, STICK

i nie tylko...

W KYAN PASCALU

Jakub CEBULA

Przemysław OSOCHA

Na

dyskietce
systemowej Kyan
Pascala brakuje
kilku bardzo
pożytecznych
procedur i funkcji.
Staraliśmy się tę
lukę wypełnić,
a efekt
prezentujemy
czytelnikom,
mając nadzieję,
że ułatwi im pracę.

Sprawa instrukcji **peek** i **poke** była już w „Bajtku” opisywana, ale zaproponowane rozwiązanie nie działa w Kyan Pascalu 2.0. W tej wersji Pascala zrezygnowano bowiem z instrukcji **assign**, na której oparta była jego konstrukcja. Postanowiliśmy więc opracować ponownie stosowne instrukcje uzupełniając je przy okazji o kilka dodatkowych.

Oto lista zdefiniowanych procedur i funkcji:

procedure poke (gdzie:real, co:integer);
czyli POKE (gdzie, co) — umieszcza w komórce pamięci o adresie 'gdzie' wartość 'co' z zakresu od 0 do 255.

procedure dpoke (gdzie, co:real);
czyli DPOKE (gdzie, co) — umieszcza w komórce pamięci 'gdzie' wartość 'co' z zakresu od 0 do 65535.

function peek (gdzie:real):integer;
czyli PEEK (gdzie) — zwraca liczbę typu integer stanowiącą zawartość komórki pamięci o adresie gdzie.

function dpeek (gdzie:real):real;
czyli DPEEK (gdzie) — zwraca liczbę typu real stanowiącą zawartość dwóch kolejnych komórek pamięci, z których pierwsza ma adres 'gdzie'.

function vstick (port:integer):integer;
czyli VSTICK (port) — zwraca wartość określającą położenie w płaszczyźnie pionowej joysticka o numerze 'port'.

function hstick (port:integer):integer;
czyli HSTICK (port) — jak wyżej, ale w płaszczyźnie poziomej.

function stick (port:integer):integer;
czyli STICK (port) — zwraca liczbę określającą położenie joysticka przyłączonego do gniazda 'port' (0 lub 1).

function strig (port:integer):integer;
czyli STRIG (port) — zwraca liczbę wskazującą wciśnięcie przycisku joysticka (0 — wciśnięty) w 'porcie' 0 lub 1.

Proponujemy stworzyć z tych procedur i funkcji jeden zestaw włączany do własnych programów przez dyrektywę **include**. Należy więc po zgłoszeniu się systemu napisać:

ED AIO.PAS

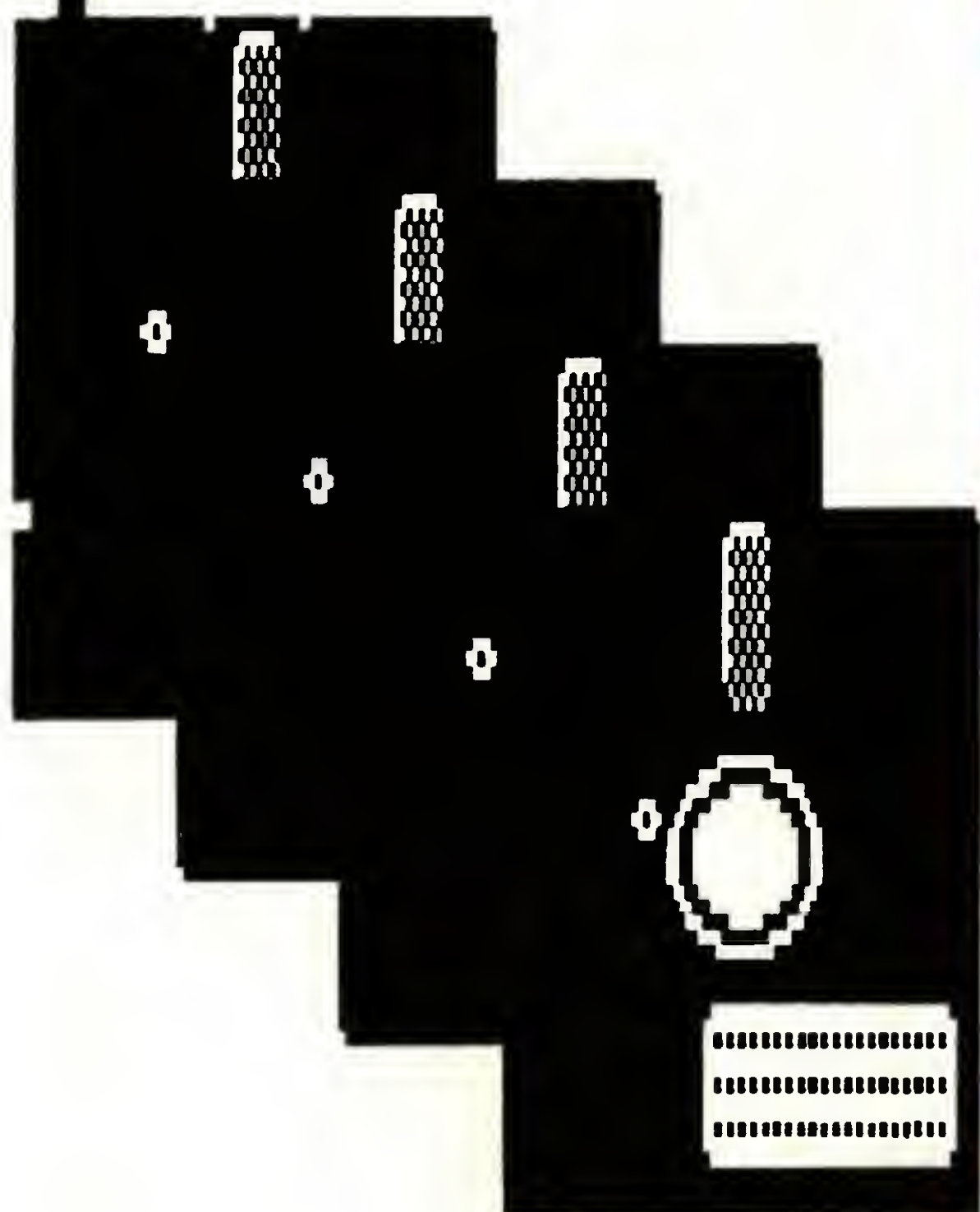
a następnie wpisać załączony listing. We własnym programie trzeba po deklaracji zmiennych umieścić dyrektywę:

#I AIO.PAS

Procedury i funkcje wywołuje się zgodnie z podanym opisem.

S
Y
S
T
E
M
Y

instalowane
w
stacjach
dysków



SPEEDY 1050

Producent: Compy Shop, RFN
Procesor: 65C02
ROM: 8 KB
RAM: 8 KB-6264
PIO: 6532
Kontroler: 2793
Transmisja: 96000 bodów
Gęstości: S, E, M, D

Oprogramowanie systemowe:

MasterDisk zawierający: program do kopiowania zabezpieczonych dyskietek w pojedynczej gęstości, program kopiujący dyskietki w gęstościach S, E, D (HSS), inicjalizator do tworzenia menu dyskietki (SPEED INIT). Ponadto wersja MasterDisk 1.2 zawiera edytor ścieżek pozwalający na tworzenie dowolnego formatu, program rozpoznający sektory podwójne i zapisane słabymi bitami (WEAK, PHANTOM, PHAZED) oraz program pozwalający na skopiowanie do bufora wybranych odcinków ze ścieżki. Wersja 1.2 pracuje w trzech prędkościach: Normal — 19200 bodów, Speedy Normal — ok. 60000 bodów i High Speedy — ok. 78000 bodów. Prędkość Normal można uzyskać zarówno programowo, jak i sprzętowo, stacja pracuje wtedy w trybie 1050 z wyłączonym buforowaniem. Wersja 1.4 posiada dodatkową prędkość Speedy Super — 96000 bodów. Pozwala to np. na formatowanie dyskietki w ciągu 9 sekund. W wersji „D” w ROM-ie stacji instalowany jest BIBODOS, a w wersji „S” — HSS Copier. Wersja „T” posiada wyświetlacz LED informujący o numerze ścieżki, błędzie itp. Oczywiście możliwe jest instalowanie wersji „TS” i „TD”.

Oprogramowanie dodatkowe:

W szybkiej wersji pracują DOS-y przystosowane do prędkości US, np. SPARTADOS, SUPERDOS 4.3E i BIBODOS 5.xF oraz wszystkie programy po zainstalowaniu w pamięci RAM stacji wariantu SPEEDY lub HAPPY.

Zalety systemu:

SPEEDY 1050 jest najszybszym systemem z dostępnych Polsce (szybszy jest — jak dotąd — jedynie I/S PLATE instalowany w Wielkiej Brytanii) całkowicie zgodnym ze wszystkimi programami napisanymi dla stacji 1050.

Wady systemu:

Ponieważ w systemie jest zainstalowany procesor 65C02, cena rozszerzenia jest bardzo duża.

HAPPY WARP
7.10

Inne nazwy: HAPPY DRIVE, Amerykan, duży HAPPY
Producent: Happy Computers Inc., USA
Procesor: 6502
ROM: 8 KB
RAM: 8 KB-6264
PIO: 6532
Kontroler: 2793
Transmisja: 70 000 bodów
Gęstości: S, E, M, D, I, A

Oprogramowanie systemowe:

Na stronie A dyskietki systemowej znajdują się liczne programy wybierane z prostego menu. Są to: 1) program do konfiguracji stacji dysków, 2) program kontrolujący poprawność ROM, RAM i obu silników stacji, 3) program kopiujący dyskietki niezabezpieczone z wykorzystaniem ramdisku, 4) program kopiujący dyskietki zabezpieczone zapisem synchronicznym (SKEW) oraz sektorami BAD, DDM, CRC w pojedynczej gęstości, 5) specjalistyczny program do kompresji (archiwizacji) plików — odtwarzanie możliwe tylko w stacjach z modulem HAPPY (nie mylić z TURBO 1050), 6) program uruchamiający tryb „H” (kolejność \$48) w systemach dołączonych stacji dysków z modulem HAPPY, 7) programy do śledzenia pracy stacji dysków (SECTOR TRACER).

Druga strona dyskietki zawiera: opis działania stacji z systemem HAPPY WARP 7.10 (w języku angielskim), program drukujący spis plików PDB znajdujących się na dyskietce, opis programu do konwersji plików (IBM-Atari i odwrotnie), specjalny WARP-SPEED DOS pracujący w gęstościach S i D, program do konwersji plików IBM FILE CONVERTER.

Oprogramowanie dodatkowe:

HAPPY FORMATER — dyskietka zawiera szereg programów, które uzupełniają dysk systemowy. Są to: FORMATER — zestaw programów do redagowania, formatowania, testowania i programowania sektorów, ścieżek oraz stacji dysków, ponadto warianty umożliwiające wczytanie do bufora całej ścieżki (łącznie z nagłówkami sektorów) i przeglądanie jej w wersji normalnej i negowanej; SEKTOR KILLER — edytor sektorowy (HEX lub ASCII); DISK MAPPER — program do sporządzania mapy dyskietki (alokacji sektorów) z zaznaczeniem wszystkich błędów (o ile występują).

I.S. PLATE

Inna nazwa: GREMLIN GRABBER
Producent: Innovated Software — Anglia
Procesor: 65C02
ROM: 64 KB
RAM: 62128-16 KB
Transmisja: 118000 bodów
Gęstości: S, E, M, D

Zestaw zawierający sześć różnych systemów zainstalowanych wewnątrz stacji. Dzięki zastosowaniu nowych specjalnych komend SIO uzyskano sześciokrotne przyspieszenie transmisji pomiędzy stacją i komputerem. Program obsługi kontrolera stacji pozwolił — poza zwiększeniem prędkości — na redukcję szumów silnika krokowego głowicy. W ROM stacji znajdują się następujące systemy:

- 1 — IS/US DOUBLER
- 2 — STANDARD 1050
- 3 — ARCHIVER
- 4 — HAPPY WARP
- 5 — LAZER
- 6 — WARP DRIVE

System jest w pełni kompatybilny z HAPPY WARP 7.xx, uruchamia się z każdym DOS-em jednak prędkość najwyższą uzyskuje ze SpartaDOS, HAPPY WARP, SPEED DOS i SUPER DOS 4.3E. Praca w gęstościach SEMD (pojedynczej, rozszerzonej, średniej i podwójnej) oraz szybkość, jaką można uzyskać w tej stacji, są jedynymi efektami powyższego zestawu. Do stacji jest dołączona dyskietka systemowa zawierająca HHSPEED menu, tester prędkości, program kopiujący sektory i specjalny program do kopiowania zabezpieczonych firmowo dyskietek. Podręcznik IS PLATE na 30 stronach opisuje instrukcje, dzięki którym użytkownik może efektywnie wykorzystać system stacji z poziomu języka maszynowego do swoich celów (dostęp do ID ścieżki, tworzenie na jednym dysku ścieżek w różnych gęstościach, własne sposoby zabezpieczania przed kopiowaniem).

Wady systemu:

Cena rozszerzenia (ok. 160 dolarów) nie jest warta efektów uzyskiwanych przez stację.

**Zbigniew
GRADOMSKI**

RAMDISK 130XE

Dla posiadaczy 130XE z magnetofonem bardzo użytecznym programem jest ramdisk z „IKS-a” 1/87.

Umożliwia on łączenie programów zapisanych przez CSAVE, zastępuje kasownik, pozwala sprawdzić poprawność nagrania, wreszcie przy jego pomocy z „Edytora Basica” mogą korzystać nie tylko osoby o anielskiej cierpliwości.

Oryginalny ramdisk zapisuje program w różnych zakamarkach pamięci. Mając Atari 130XE postanowiłem wykorzystać dodatkową pamięć do stworzenia nowego ramdysku. Od wzorca mój program różni się tym, że jest krótszy (usunąłem urządzenie „N:”), ale ma kilka dodatkowych zalet.

Po wpisaniu procedury obsługi ramdysku na szóstą stronę pamięci jego uruchomienie następuje przez **A=USR (1560)**. Programy zapisuje się w nim instrukcjami **SAVE "B:"** lub **LIST "B:"**, odczytuje zaś odpowiednio poprzez **LOAD "B:"** lub **ENTER "B:"**. Instrukcję **A=USR (1560)** należy zastosować również po naciśnięciu klawisza «RESET».

Z zainstalowanym ramdyskiem proponuję wykonać kilka

doświadczeń. Proszę wgrać do ramdysku dowolny program i napisać w trybie bezpośrednim:

```
POKE 766, 1:POKE 54017, 201:F, 1=16384:TO 200
OO: ?CHR$(PEEK(1)):N, 1
```

Wykonując to doświadczenie dla programów zapisanych przez SAVE i LIST będziemy mogli zobaczyć, jak wygląda nagrany program. Teraz sprawdzimy i zapamiętajmy zawartość komórek 1683 i 1684, po czym pożegnajmy się z naszym programem przez **BYE** lub **A=USR(58487)**. Pamięć RAM komputera zostaje wyczyszczona, ale tylko pamięć podstawowa, a nie dodatkowa! Ponownie wgrywamy i uruchamiamy ramdisk, do komórek 1683 i 1684 wpisujemy zapamiętane wartości i... odzyskujemy nasz program. Jeśli przejście do zimnego startu nie było wynikiem eksperymentu, lecz „wypadkiem przy pracy”, znajomość zawartości komórek 1683 i 1684 nie jest niezbędna. Można tam wpisać 255 i 127, a pracowite poszukiwanie końca programu przerwać komputerowi klawiszem «RESET».

Jeżeli nie odpowiada nam założenie ramdysku w trzecim banku pamięci dodatkowej, możemy w wierszu 46 wpisać w DATA wartości 193, 197 lub 205 zamiast 201. Wartości te można również wpisać do komórki 1666 uruchomionego programu. W ten sposób można założyć CZTERY niezależne ramdyski!

Marek OMIRSKI

```
GP 1 REM RAMDISK 130XE
ON 2 REM Marek Omirski
DV 3 REM (c) 1991, Sp. Bajtek
NJ 4 REM
BI 10 S=0:FOR I=1536 TO 1684:READ A:S=S+A
:POKE I,A:NEXT I
HZ 12 IF S<>14833 THEN ? "Popraw dane!":E
ND
AW 14 A=USR(1560):REM Powtorz po RESET
GV 20 DATA 162,66,160,28,169,6,32,188,238
,144
QH 22 DATA 12,192,255,240,8,157,27,3,152,
157
SR 24 DATA 26,3,24,96,104,76,0,6,42,6
UI 26 DATA 50,6,83,6,53,6,50,6,50,6
TO 28 DATA 76,51,6,169,64,133,75,169,0,13
3
CI 30 DATA 74,160,1,96,160,0,32,122,6,145
SU 32 DATA 74,32,137,6,230,74,196,74,208,
2
XP 34 DATA 230,75,200,165,74,141,147,6,16
5,75
CR 36 DATA 141,148,6,96,172,147,6,174,148
,6
TU 38 DATA 228,75,208,7,196,74,208,3,160,
136
NB 40 DATA 96,160,0,32,122,6,177,74,32,13
7
LB 42 DATA 6,230,74,196,74,208,2,230,75,1
60
KE 44 DATA 1,96,120,140,14,212,174,1,211,
160
FL 46 DATA 201,140,1,211,160,0,96,142,1,2
11
CO 48 DATA 162,255,142,14,212,88,96,0,64
```

PROGRAMOWANIE STACJI

TOMS MULTI

Rozszerzenie TOMS MULTI, którego test był opublikowany w „Bajtku” 5–6/90, jest jedną z najpopularniejszych modyfikacji stacji dysków LDW 2000 i CA-2001. Jednak niewielu użytkowników wie, że jej praca może być programowana.

Stacja dysków MULTI ma częściowo otwartą strukturę programową, umożliwiającą wykonywanie niestandardowych funkcji, na etapie pisania systemu operacyjnego stacji jeszcze nieokreślonych, lub w ogóle nieprzewidywanych. Częściowo otwarta struktura oznacza, że zostały ustalone pewne sztywne ramy ogólne, którym musi odpowiadać rozszerzenie programowe stacji, jednakże w tych ramach panuje dość duża dowolność, i sama istota rozszerzenia programowego nie jest przez system operacyjny stacji kontrolowana. Po uruchomieniu rozszerzenia programowego, choćby jednokrotnym, na programistę spada całkowita odpowiedzialność za dalsze prawidłowe funkcjonowanie stacji. Pewność całkowitego usunięcia skutków rozszerzenia daje jedynie wyłączenie zasilania stacji. Jest to istotne o tyle, że uszkodzenie mechaniczne bądź elektryczne stacji na drodze programowania jej, choć trudne do wykonania, to jednak nie jest wykluczone. Programista piszący rozszerzenie programowe stacji musi wiedzieć, co robi.

Do ładowania rozszerzeń do stacji i ich wykonywania służy komenda "X". Komendzie tej towarzyszy bufor danych o długości 256 bajtów (przy nadawaniu do stacji). Przy odbiorze długość bufora nie jest określona, i zależy od ustaleń poczynionych na etapie pisania rozszerzenia, choć jest przewidziany uproszczony mechanizm nadawania danych ze stacji, jeśli długość bufora wynosi także 256 bajtów. W praktyce do tej pory były stosowane długości od 0 (brak danych przy odbiorze) do 8 KB (np. odczyt całej ścieżki).

Ponieważ do transmisji w obie strony wykorzystywana jest jedna tylko komenda X, to muszą być jakieś mechanizmy pomocnicze, za pomocą których stacja rozpoznaje, o jaką ewentualność chodzi komputerowi. Tymi znacznikami są bity 4, 5 i 6 bajtu AUX2 nadawanej komendy. Bit 7 nie jest przy tym wykorzystywany (przy wywołaniu SIO powinien być skasowany), gdyż kolidowałby z systemem szybkiej transmisji typu TURBO. Jeśli bit 6 jest ustawiony, to stacja odbiera dane od komputera (zawsze 256 bajtów). Jeśli przy tym bit 5

LISTING 1

```
***** LISTING 1 *****
;komenda ładuje albo wykonuje moduły zewnętrzne.
;Adres ładowania i startu jest ALTBUF 7800+700.

XCMD: CALL ACKCDM ;zaackuj
      CALL BUF0F ;wylacz buforowanie
      LD A,(AUX2) ;odbior czy nadawanie?
      BIT 6,A
      JR 2,XNADAW ;lezy to wykonaj i nadaj dane
      LD HL,ALTBUF ;bufor danych
      BIT 5,A
      JR NZ,PROG
      LD HL,(DATVEC)
      LD BC,100 ;zawsze 256 bajtow
      CALL RECVHL ;i odebrac
      JP NZ,DATERR ;skocz gdy blad
      CALL ACKDAT ;zaackuj
      LD A,(AUX2)
      BIT 4,A
      JR 2,XOK
      CALL ALTBUF
XEND: CALL BIOUT
      XOR A
      RET
XOK: LD A,'C'
      JR XEND

XNADAW:
      CALL ALTBUF ;tu wykonaj
      LD BC,100
      LD HL,(DATVEC)
      CALL SENDHL
      XOR A
      RET

***** KONIEC LISTINGU *****
```

jest ustawiony, to do stacji przesyłany jest program, ładowany do bufora programu, a jeśli bit ten jest skasowany, to ładowane są dane (do bufora danych). Bufor danych nie ma stałego adresu, lecz jest wskazywany przez wektor DATVEC. Przy odbiorze przez stację od komputera istotny jest także bit 4. Jeśli jest on ustawiony, to po skończonej transmisji 256-bajtowego bufora zostanie wywołany program aktualnie znajdujący się w buforze rozszerzenia. Jeśli bit 4 jest skasowany, to transmisja zostanie zakończona bez wywołania rozszerzenia.

Jeśli bit 6 bajtu AUX2 jest skasowany, to komenda w zasadzie jest związana z transmisją do komputera. Bezpośrednio po odbiorze komendy wywoływane jest aktualne rozszerzenie, może więc ono także odbierać dane (w tym wypadku o dowolnej długości). Jeśli rozszerzenie zamierza nadawać do komputera, to gdy umieści swoje dane w buforze wskazywanym przez DATVEC i liczba danych wynosi 256 B, wystarczy wrócić do obsługi komendy "X" z kodem błędu w akumulatorze, by program realizujący komendę X sam zatroszczył się o transmisję. Jeżeli rozszerzenie chce odbierać, lub nadawać inną liczbę bajtów, to obsługuje transmisję samo.

Dla wyjaśnienia wątpliwości na listingu 1 pokazany jest ORYGINALNY fragment programu źródłowego ROM stacji, który realizuje obsługę komendy "X". Listing nie był przeznaczony do publikacji, dlatego komentarze są trochę „tajemnicze”, miały służyć bowiem tylko autorowi programu.

Dla użytku rozszerzeń programowych ustalono pewne — niezmiennie w różnych modyfikacjach programu stacji — stałe: adresy komórek RAM i wektory wejścia do procedur. Stałe są używane w postaci tekstu przyłączanego przy kompilacji do tekstu rozszerzenia dyrektywą INCLUDE. Tekst ten nosi nazwę DEFS.MAC. Nie wszystkie zawarte w nim definicje są istotne dla rozszerzeń, a tylko te, które odnoszą się do sprzętowych adresów portów na płycie elektroniki i podziału przestrzeni adresowej procesora. Listing 2 zawiera prawie cały tekst DEFS.MAC, z wyjątkiem krótkiego fragmentu związanego z ochroną programu przed fałszerstwami (zresztą niezbyt rozbudowaną i już przez złodziei pokonaną).

Adresy RAM i ROM dostępne dla rozszerzeń są zadeklarowane w programie jako globalne i dostępne są dla rozszerzenia dopiero w fazie konsolidacji, poprzez przyłączenie modułu BAZA.REL. Taka metoda jest wygodna dla autora systemu, dysponującego jego tekstem i modułem relokacyjnym. Tekst systemu i moduł .REL nie są prze-

LISTING 2

```
***** LISTING DEFS.MAC *****

.Z80
.RADIX 16

; Fragment poufny, wycięte.

;deklaracje statycznych
;definicje komend dla sterownika dyskowego WD 2797

WDREAD EQU 88 ;read sector
WDWRIT EQU 0A8 ;write sector
WDADDR EQU 0C0 ;read address
WDFORM EQU 0F0 ;write track

;gestosc na jaka stacja ustawia sie bez dyskietki
DEFMOD EQU 2 ;default mode ED

;ilosc zliczen okreslajaca czestotliwosc migania
;lewego wyswietlacza

TOGDEL EQU OFF ;wartosc licznika migania lewego

;deklaracje bitow w latch'ach
;maski bitowe wycinajace z portu ATARI bity odpowia-
;dajace poszczegolnym sygnałom zlacza Atari
;i sygnałom od WD do komputera

CLKIN EQU 01 ;port ATARI.
;linie transmisji do komputera

CLKOUT EQU 02
DIN EQU 04
DOUT EQU 08
RDY EQU 10
CMD EQU 20
DRQ EQU 40 ;noga DRQ WD 2797
IRQ EQU 80 ;noga IRQ WD 2797

DIP0 EQU 01 ;port KEYCLR, maski dip switch
DIP1 EQU 02
SINGLE EQU 04

TRKKEY EQU 10 ;maski poszczegolnych klawiszy
TYPKEY EQU 20
ERRKEY EQU 40
CHANGE EQU 80 ;protect albo zmiana dysku
ALLKEY EQU 70 ;maska na wszystkie klawisze naraz

BUSLED EQU 80 ;port LEDY, dioda Busy
ENP EQU 80 ;port LEDENP, noga ENP WD 2797

;definicje wartosci do wpisania w zatrzask wyswietlacz
;aby uzyskac czytelne znaki

Y0 EQU 0C0 ;cyferki 0 - A
Y1 EQU 0F9
Y2 EQU 0A4
Y3 EQU 0B0 ; numery bitow
Y4 EQU 99 ; 0
Y5 EQU 92 ; 5 1
Y6 EQU 82 ; 6 wzor znaku
Y7 EQU 0F8 ; 4 2
Y8 EQU 80 ; 3
Y9 EQU 90
YA EQU 88
YB EQU 83
YC EQU 0C6
YD EQU 0A1
YE EQU 86
YF EQU 8E
Y1 EQU 0CF
YM EQU 0BF ;minus
YP EQU 8C
YS EQU Y5
YG EQU 0C2
YH EQU 89
YL EQU 0C7
YU EQU 0C1

;deklaracje portow I/O

BEEPPI EQU 00 ;linia Sound w +
BEEPL0 EQU 01 ;linia Sound w -
JUMPL0 EQU 02 ;Extra jumper
JUMPHI EQU 03
DOUTHI EQU 04 ;Data Out (do komputera)
DOUTLO EQU 05
COUTHI EQU 06 ;Clock Out
COUTLO EQU 07
DDENON EQU 08 ;noga DDEN kosci WD
DDENOF EQU 09
MOTOF EQU 0A ;włącznik silnika
MOTON EQU 0B
IPUP EQU 0C ;noga IP (index pulse) WD
IPDOWN EQU 0D
WRITON EQU 0E ;daje 1 na nóżce 4 LS02, puszcza WG
WRITOF EQU 0F ;daje 0, blokada WG

;deklaracje dekodowania pamieci

KEYCLR EQU 1000 ;odczyt klawiszy z ich skasowaniem
KEYREA EQU 1001 ;odczyt klawiszy bez kasowania
ATARI EQU 2000 ;port zlacza do komputera
STEPPER EQU 3000 ;zatrzask faz silnika krokowego
LEDY EQU 4FFF ;prawy wyswietlacz (albo oba)
```

znaczone do rozpowszechniania, dlatego dla osób próbujących pisać rozszerzenia praktyczniejsze będzie podanie wartości etykiet globalnych wprost, w formie fragmentu wydruku assemblera (listing 3).

Wyjaśnienie znaczenia poszczególnych etykiet podano na listingu 4, który jest fragmentem tekstu źródłowego programu stacji. Nie jest to jednak oryginalny fragment programu źródłowego, gdyż dopisano w nim komentarze dla czytelnika, wyjaśniające bliżej przeznaczenie procedur i komórek pamięci.

Praktyczne przykłady pisania rozszerzeń programowych stacji podane są na listingu 5 — są to procedury pisane do programu kopiującego TDK, programu śledzącego (tracera) i inne. Użytkownik pisząc własne procedury może wzorować się na podanych przykładach, nie powinien jednak uży-

LISTING 3

```
*** WYDRUK WARTOSCI STALYCH GLOBALNYCH ***

ADRBUF 78041
ALTBUF 7F001
AUX1 78021
AUX2 78031
CMDBUF 7C001
COMBUF 78001
DATBUF 7A001
DATVEC 780A1
GAPLEN 78601
HIGAP 78611
INSTAT 781B1
LEN 78071
PRZVEC 78181
SEC 78061
SID 78051
TRACK2 781C1
TRK 78041
VACKDA OFEA1
VB1OUT OFED1
VBELL OFDB1
VCZTAD OFC31
VCZYTW OFD21
VDATER OFE71
VDSEEK OFC91
VFORTR OFD51
VINPIS OFDE1
VMOTGO OFCF1
VREAAD OFDB1
VRECVH OFE41
VSEEKO OFC61
VSENDH OFE11
VWAIIR OFCC1
VWDCLR OFC01
WDSTAT 781A1

*** KONIEC WYDRUKU GLOBALNYCH STALYCH KOMPILACJI ***
```

wać dyrektywy .REQUEST oraz nie powinien stosować stałych globalnych (identyfikowanych przez # #) — stałe takie muszą być zdefiniowane jawnie jako EQU korzystając z adresów podanych wcześniej. Składnia dostosowana jest do assemblera M80 i linkera L80. Podane listingi są uzupełniane komentarzami dla czytelnika, których w działających wersjach nie było. Podane nazwy programów w znacznym stopniu odpowiadają nazwom, jakie mają bloki cyfr, używane w programach dla Atari (wiersze DATA w Basicu lub DB w assemblerze). Dokładne informacje o sposobie użycia tych — gotowych już — procedur, bez pisania własnych, można znaleźć, analizując ogólnie dostępne listingi programów na Atari, wykorzystujących rozszerzenia programowe.

Napisane rozszerzenie należy skompilować (np. programem M80) i skonsolidować (np. linkerem L80). Z tak powstałego pliku .COM należy odpowiedni fragment zamienić na instrukcje DATA lub DB i włączyć do programu dla Atari, który posługując się wywołaniami SIO dokona transmisji rozszerzenia do stacji i odpowiednio wykorzysta wyniki. Przykłady mogą tu być programy usługowe do MULTI, napisane w Turbo Basicu, jak GETTRACK, PUTTRACK, INDEX FORMATTER itd.

Robert MIZIŃSKI

LISTING 4

```
***** FRAGMENTY Z DYREKTYWAMI GLOBAL *****

;Program ROM: wektory prowadzące do procedur.
;Od adresu F00 zaczyna sie tablica skokow.

JMPTAB EQU OFC0

VWDCLR: JP WDCLR ;zerowanie kosci sterownika FDD
;wraca ze statusem WD z momentu
;przed zerowaniem

VCZTAD: JP CZYTAD ;odczytanie pierwszego napotkanego
;adresu sektora do bufora wskazy-
;wanego przez HL, bez ustawiania
;licznika oczekiwania w DE powrot
;NZ to czas uplynal

VSEEKO: JP SEEKO ;jazda wozkiem na sciezke 0
VDSEEK: JP DSEEK ;jazda wozkiem na sciezke podana
;w rejestrze D, numer sciezki jest
;pomnozony przez 2

VWAIIR: JP WAIIRQ ;wraca ze statusem WD po IRQ
VMOTGO: JP MOTGO ;motor napędu dyskowego włącz,
;jeśli sie kreci to wraca natychmiast,
;inaczej czeka az sie rozkreci

VCZYTW: JP CZYTWD ;czytaj z WD do bufora podanego
;w DE, w BC liczba bajtow do
;odczytania. Jeśli czas uplynie,
;to wraca 2, jeśli sie uda, to NZ
;formatuj jedna sciezke

VFORTR: JP FORTRK ;czyta adres do ADRBUF, ustawiajac
VREAAD: JP READAD ;sobie czas w DE na 4000H (licznik
;oczekiwania na przeczytanie
;jakiegos adresu)
;zadzwon dzwonkiem stacji
VINPIS: JP INPISZ ;pisanie do WD z bufora HL, liczba
;bajtow w BC, w A komenda dla WD.
;Wraca 2 jeśli czas uplynal
VSENDH: JP SENDHL ;wyslanie do komputera BC bajtow
;z bufora HL, najpierw posyłany
;jest kod bledu z A
```

```

VRECVH: JP RECVHL ;odbior od komputera BC bajtow
;do HL. Sprawdza sume kontrolna
;i wraca NZ, jesli jest zla.
VDATER: JP DATERR ;zaznacza w wewnetrznym statusie
;stacji, ze byl blad odbioru
;ramki danych
VACKDA: JP ACKDAT ;poslanie potwierdzenia po ramce
;danych ACK ("A") z odpowiednim
;opoznieniem
VBIOUT: JP BIOUT ;poslanie jednego bajtu do komputera
;z A z opoznieniem

```

```

GLOBAL VWDCLR,VCZTAD,VSEEEK,VSEEEK,VWAIIR,VHOTGO
GLOBAL VCZYTW,VFORTR,VREAAD,VBELL,VINPIS
GLOBAL VSENDH,VRECVH,VDATER,VACKDA,VBIOUT

```

Adresy komorek pamieci RAM:

```

ORG RAM
COMBUF: DS 2 ;bufor komend do stacji
AUX1: DS 1 ;bajty pomocnicze komend
AUX2: DS 1
GLOBAL AUX1,AUX2
ADRBUF: ;bufor adresow odczytywanych z dysku
TRK: DS 1 ;tu wchodzi numer sciezki
SID: DS 1 ;tu strona
SEC: DS 1 ;numer sektora
LEN: DS 1 ;kod dlugosci sektora
DS 2 ;miejsce na sume kontrolna
DATVEC: DS 2 ;przechowuje adres bufora danych
GLOBAL COMBUF,ADRBUF,DATVEC,TRK,SID,SEC,LEN

PRZVEC: DS 2 ;wektor na inny niz standardowy
;przeplot sektorow przy formatowaniu

GLOBAL PRZVEC
WDSTAT: DS 1 ;ostatni status WD
GLOBAL WDSTAT
INSTAT: DS 1 ;bitowy wewnetrzny status stacji
;definicje znaczenia bitow:
;0 - 1 to command frame error
;1 - 1 to bad data frame received
;2 - 1 to FDC error
;3 -
;4 - 0 to motor off, 1 to on
;5 - 1 to DD
;6 - 1 to ID
;7 - 1 to ED
GLOBAL INSTAT

```

```

TRACK2: DS 1 ;track obliczony z komendy, *2
GLOBAL TRACK2

GAPLEN: DS 1 ;dlugosc przerwy miedzy sektorami
;przy formatowaniu
HIGAP: DS 1 ;starszy bajt tejze przerwy
GLOBAL GAPLEN,HIGAP

```

```

DATBUF EQU RAM+200 ;standardowy bufor na dane
CHDBUF EQU RAM+400 ;bufor na komende do stacji
ALTBUF EQU RAM+700 ;bufor do modulu zewnetrznych
GLOBAL DATBUF,CHDBUF,ALTBUF

```

*** KONIEC FRAGMENTOW LISTINGU Z DYREKTYWAMI GLOBAL ***

```

LEDENP EQU 5000 ;lewy wyswietlacz
CRSR EQU 6000 ;WD 2797 command/statut register
TR EQU 6001 ;track register
SR EQU 6002 ;sector register
DR EQU 6003 ;data register
RAM EQU 7800 ;poczatek pamieci RAM (o dlugosci 10 KB)
EXRAM EQU 8000 ;extra ram (koncowe 8 KB)
EXROM EQU 0A000 ;extra rom
KOPPRG EQU 0A400 ;poczatek programow bootowalnych
DIRSEC EQU 0FF80 ;ostatnie 128 bajtow rom zawiera
;sektor 361, pierwszy katalogu

```

***** KONIEC DEFS.MAC *****

LISTING 5

***** LISTINGI ROZSZERZEN UZYWANYCH PRZEZ AUTORA *****

NAZWA seek
program jedzie wzkiem glowicy na sciezke przekazana
jako parametr w AUX1, numer sciezki jest pomnozony *2

```

.Z80
ASEG
.RADIX 16
ORG 1100
.PHASE 7800 + 700

INCLUDE DEFS

CALL VWDCLR;; reset WD
CALL VMOTGO;; start silnika
LD A,(AUX1); jest tam numer sciezki razy 2
.RADIX 10
CP 40*2 ;sprawdz,czy nie za duzy
.RADIX 16
JR NZ,ZLE ;jesli za duzy, to nie rob
LD D,A ;tu zaladuj parametr
CALL VDSEEEK;; wywołanie procedury ruchu wzkiem
LD A,'C' ;powrot COMPLETED
RET
ZLE: LD A,'E' ;tu powrot ERROR
RET

.DEPHASE
END

```

NAZWA wielopis
program zapisuje sektor wielokrotnie, wystepujacy na
sciezce wiecej niz raz, metoda to wpierv dopasowanie
sie do podanego dalej przeplotu, w postaci ciagu
numerow sektorow, poprzez odczytanie pod rzad
wszystkich adresow z listy, a potem zapisanie sektora
o wlasnym numerze. tresc sektora jest pod DATVEC

```

.Z80
ASEG
.RADIX 16
ORG 1100
.PHASE 7800 + 700

INCLUDE DEFS

ILSTAR EQU *0C0 ;ile sektorow przeczytac na starcie,
do
;dobasowania przeplotu
TRACK EQU *0C1 ;numer sciezki
DLUG EQU *0C2 ;dlugosc sektorow

LISTAR EQU *0C4 ;lista numerow sektorow

CALL VWDCLR;; reset sterownika
CALL VMOTGO;; start silnika
LD C,30 ;30 prob zlapania synchronizacji
LD HL,LSTAR ;wyceluj na liste numerow sektorow
CALL ADR
JR NZ,AGA
LD A,(ILSTAR)
LD B,A
INC HL
CIAG: CALL ADR ;byc moze juz idzie po liscie
JR Z,WCIAGU
DOKONC: CALL ADR
DJNZ DOKONC
DEC C

```

```

JR NZ,AGA
JR ZLE
WCIAGU: INC HL
DJNZ CIAG

LD A,(TRACK) ;zaladuj rejestry WD
LD (TR),A
LD A,(HL)
LD (SR),A
LD HL,(DATVEC);
LD BC,(DLUG)
LD A,(AUX2); ;sformuj komende dla WD
;odpowiednio do typu
;zabezpieczenia sektora

AND 9
OR OAO
LD DE,ATARI
LD (CRSR),A
DRQ?: LD A,(DE) ;czekaj na pierwsze DRQ
BIT 6,A
JR NZ,ISDRQ
BIT 7,A
JR Z,DRQ?
JR SKONC

ISDRQ: LD A,(HL) ;zapisuj kolejne bajty
CPL
LD (DR),A
INC HL
DJNZ DRQ?

SKONC: LD A,(AUX2); ;sprawdz zabezpieczenie, konczyc
;normalnie czy zrobic CRC error

AND 8
JR NZ,ANOCRC

LD B,2
LD A,(DE)
AND DRQ
JR Z,DRQ2
LD (DR),A
DJNZ DRQ2
CALL VWDCLR;;
JR DOBZE

ANOCRC: CALL VWAIIR;; ;tu koniec normalny

DOBZE: LD A,'C' ;zwroc COMPLETED
RET
ZLE: LD A,'E' ;zwroc do komputera blad
RET

ADR: PUSH BC ;sproboj odczytac adres pierwszego
;sektora z listy
PUSH HL
CALL VREAAD;; ;przeczytaj adres
JR NZ,NIEZLE
POP HL ;blad czasu, nic nie czyta
POP HL ;zwrot bledu do komputera
POP HL
JR ZLE

NIEZLE: CALL VWAIIR;; ;odczeka ukonczenia komendy
AND 08 ;czy bylo CRC ok?
JR Z,NOCRC
CALL VWDCLR;; ;zle CRC, sproboj jeszcze
JR AD1
POP HL
POP BC
LD A,(SEC); ;pobierz numer sektora z adresu
CP (HL) ;sprawdz z lista
RET ;wroc z wynikiem porownania

.DEPHASE
END

```

NAZWA tylkoad
program odczytuje serie adresow sektorow ze sciezki,
na ktorej znajduje sie glowica. Zapamietywane sa tylko
numery sektorow. Do komputera odsylana jest lista
odczytanych numerow, pierwsza jej elementem jest
liczba odczytanych prawidlowo numerow. Czas, w ciagu
ktorego nastepuja odczyty, jest wpisany jako parametr,
podczas przesyłania tego rozszerzenia do stacji.
Sluzy do ustalenia przeplotu formatowania sciezki

```

.Z80
ASEG
.RADIX 16
ORG 1100
.PHASE 7800 + 700

CZAS EQU *+80
BUFF EQU *+82 ;tu miejsce na odczytany jeden adres
CALL VWDCLR;;
CALL VMOTGO;;
LD DE,(CZAS) ;tam trzeba wpisac czas trwania petli
LD HL,(DATVEC); ;odczytane numery beda wpisywane
;w standardowy bufor danych
INC HL ;pierwsza pozycja omln, miejsce
;na liczbe odczytow

XOR A
PUSH AF

ZCZYT: PUSH HL
LD HL,BUFF
CALL VCZTAD;; ;odczytaj adres do mojego bufora
POP HL
JR Z,TIME ;przerwij jesli nic sie nie czyta
CALL VWAIIR;; ;CRC
AND 08
JR NZ,ZCZYT ;tylko poprawne adresy obrabiamy
POP AF
INC A
AND A
JR Z,ZADUZO
PUSH AF
LD A,(BUFF*2) ;z bufora wybierz numer sektora
LD (HL),A ;i zapisz go do bufora danych
INC HL
JR ZCZYT

TIME: POP AF
ZADUZO: LD HL,(DATVEC); ;w poczatek bufora danych
LD (HL),A ;wpisz liczbe odczytanych numerow
CALL VWDCLR;; ;na koniec oczyść kontroler
LD A,'C' ;i wroc COMPLETED
RET
ZLE: LD A,'E' ;wyjście ERROR
RET

.DEPHASE
END

```

NAZWA pelad
program odczytuje liste pelnych adresow sektorow
:(po cztery bajty) z aktualnej sciezki.
Sluzy do pelnej orientacji w obsadzeniu sciezki.

```

.Z80
ASEG
.RADIX 16
ORG 1100
.PHASE 7800 + 700

CZAS EQU *+80
CALL VWDCLR;;
CALL VMOTGO;;
LD DE,(CZAS) ;czas odczytu jest parametrem
LD HL,(DATVEC);
INC HL
XOR A ;zeruj licznik adresow
PUSH AF
ZCZYT: CALL VCZTAD;;
DEC HL
DEC HL
JR Z,TIME

```

```

CALL VWAIIR;;
AND 08 ;CRC
JR NZ,NOINC
POP AF
INC A ;zwiększ licznik
CP 40 ;40 to maksimum
JR Z,ZADUZO
PUSH AF
JR ZCZYT
NOINC: DEC HL
DEC HL
DEC HL
DEC HL
JR ZCZYT

TIME: POP AF
ZADUZO: LD HL,(DATVEC);
LD (HL),A ;liczba adresow jako pierwsza w
;buforze
CALL VWDCLR;;
LD A,'C'
RET
ZLE: LD A,'E'
RET

.DEPHASE
END

```

NAZWA czyittrak
sluzy do odczytania calej sciezki komenda Reso Track

```

.Z80
ASEG
.RADIX 16
ORG 05000
.PHASE 7800 + 700

INCLUDE DEFS

CALL VWDCLR;;
CALL VMOTGO;;

LD A,0E0 ;wbij komende
LD (CRSR),A
LD A,(AUX1); ;w auxie 1 jest czas czytania / 255
LD B,A ;BC to licznik
LD C,0
LD DE,EXRAM*2 ;adres bufora danych
LD HL,0500 ;po wbiciu komendy trzeba
;dosc dlugo poczekac
LD A,H ;zanim WD zacznie monitorowac
OR L ;noge IP/
JR NZ,DELAY
IN A,(IPDOWN) ;mrygnij index pulsem
PUSH HL ;niech troche polezy
POP HL
PUSH HL
POP HL
LD HL,ATARI
IN A,(IPUP)
TIME: DEC BC
LD A,B
OR C
JR Z,KONIEC
BIT 6,(HL) ;w HL jest ATARI
JR Z,TIME
LD A,(DR)
LD (DE),A
INC DE
JP TIME

KONIEC: EX DE,HL ;adres pierwszej wolnej komorki do HL
LD DE,EXRAM*2 ;do DE poczatek bufora
CCF
SBC HL,DE ;wylicz ilosc bajtow odczytanych
LD (EXRAM),HL ;i wpisz na poczatku bufora
CALL VWDCLR;;
LD A,'C' ;zawsze powrot bez bledu
LD BC,2000 ;caly ram dodatkowy
LD HL,EXRAM
CALL VSENDH;; ;wyslij po linii
POP HL ;zdejmij adres powrotny
XOR A ;jest OK
RET ;i wroc do petli glownej

.DEPHASE
.REQUEST BAZA
END

```

NAZWA badform
sluzy do formatowania sciezki sektorami bez zawartosci,
ktore do momentu zapisania sa bad sektorami. Metoda:
prowadzi sie zapis sciezki komenda Write Track,
zawartosc sciezki jest przygotowana w 8 KB buforze,
przy czym dla wszystkich bajtow rownych specjalnemu
znacznikowi (tu EE), kontroler jest odcinany od napadu
poprzez wyłozenie WG. Do tego wykorzystuje sie dodatkowo
wbudowany układ odcinania. Przeznaczone do wytwarzania
ipodkladu pod slabe bity

```

.Z80
ASEG
.RADIX 16
ORG 05000
.PHASE 7800 + 700

INCLUDE DEFS

OFF EQU 0EE

LD HL,EXRAM
LD BC,2000
CALL VRECVH;;
JP NZ,VDATER;;
CALL VACKDA;;
CALL VWDCLR;;
CALL VMOTGO;;
LD A,0F0 ;wbij komende
LD (CRSR),A
LD BC,(EXRAM) ;pierwsze 2 bajty bufora to licznik
LD DE,EXRAM*2 ;adres bufora danych
LD HL,ATARI
DANA1: BIT 6,(HL)
JR Z,DANA1
XOR A
LD (DR),A

PUSH HL
LD HL,0500 ;po wbiciu komendy trzeba

DELAY: DEC HL ;dosc dlugo poczekac
LD A,H ;zanim WD zacznie monitorowac
OR L ;noge IP/
JR NZ,DELAY
IN A,(IPDOWN) ;mrygnij index pulsem
POP HL
PUSH HL
POP HL
IN A,(IPUP)

TIME: BIT 6,(HL) ;w HL jest ATARI
JR Z,TIME
LD A,(DE)
LD (DR),A
CP OFF

JR NZ,DOON
IN A,(WRITOF)
JP GODALE
DOON: IN A,(WRITON)

GODALE: INC DE
DEC BC
LD A,B
OR C
JR NZ,TIME

CALL VWDCLR;;
LD A,'C' ;zawsze powrot bez bledu
CALL VBIOUT;; ;wyslij po linii
POP HL ;zdejmij adres powrotny
XOR A ;jest OK

```

```

RET          ;i wroc do petli glownej

.DEPHASE
.REQUEST BAZA
END

;NAZWA for16
; 4 lipca 89
;wycieto wlasny seek, musi wziesniej byc seek dodatkowy
;inny uklad parametrow niz w starym for.mec
;formatuje sciezke na dowolny przeplot, podany jako
;parametr po rozszerzeniu

.Z80
ASEG
.RADIX 16
ORG 1100
.PHASE 7800 + 700

INCLUDE DEFS

GAP EQU $+80
HGAP EQU $+81
PRZEP EQU $+82

CALL VWDCLR##
CALL VMOTGO##
LD HL,PRZEP
LD (PRZVECH##),HL
LD A,(GAP)
LD (GAPLEN##),A
LD A,(HGAP)
LD (HGAP##),A
CALL VFORTR##
AND 44
JR NZ,ZLE
LD A,'C'
RET
ZLE: LD A,'E'
RET

.DEPHASE
END

;NAZWA ramdump
;test pozwalajacy obejrzec zawartosc calego ramu, staoji

.Z80
ASEG
.RADIX 16
ORG 05000
.PHASE 7800 + 700

INCLUDE DEFS

LD A,'C' ;zawsze powrot bez bledu
LD BC,2800 ;caly ram zwykly+dodatkowy
LD HL,RAM
CALL VSENDH## ;wyslij po linii
POP HL ;zdejmiij adres powrotny
XOR A ;jest OK
RET ;i wroc do petli glownej

.DEPHASE
.REQUEST BAZA
END

;NAZWA puttrack
;zapisuje cala sciezke zawartoscia przeslanego bufora

.Z80
ASEG
.RADIX 16
ORG 05000
.PHASE 7800 + 700

INCLUDE DEFS

LD HL,EXRAM
LD BC,2000
CALL VRECVR##
JP NZ,VDATER##
CALL VACKDA##
CALL VWDCLR##
CALL VMOTGO##
LD A,OFO ;wbij komende
LD (CRSR),A
LD BC,(EXRAM) ;pierwsze 2 bajty bufora to licznik
LD DE,EXRAM+2 ;adres bufora danych

DANA1: BIT 6,(HL)
JR Z,DANA1
XOR A
LD (DR),A

PUSH HL
LD HL,0500 ;po wbiciu komendy trzeba
DEC HL ;dosc dlugo poczekac
LD A,H ;zanim WD zacznie monitorowac
OR L ;noge IP/
JR NZ,DELAY
IN A,(IPDOWN) ;mrygnij index pulsem
POP HL
PUSH HL
POP HL
IN A,(IPUP)

TIME: BIT 6,(HL) ;w HL jest ATARI
JR Z,TIME
LD A,(DE)
LD (DR),A
INC DE
DEC BC
LD A,B
OR C
JR NZ,TIME

CALL VWDCLR##
LD A,'C' ;zawsze powrot bez bledu
CALL VB1OUT## ;wyslij po linii
POP HL ;zdejmiij adres powrotny
XOR A ;jest OK
RET ;i wroc do petli glownej

.DEPHASE
.REQUEST BAZA
END

;NAZWA jednopis
;zapisuje sektor wystepujacy na sciezce tylko raz
;swytwarza blad odpowiednio do parametru w aux2

.Z80
ASEG
.RADIX 16
ORG 1100
.PHASE 7800 + 700

INCLUDE DEFS

DLUG EQU $+80
TRACK EQU $+83

CALL VWDCLR##
CALL VMOTGO##
LD A,(TRACK)
LD (TR),A
LD A,(AUX1##)
LD (SR),A
LD HL,(DATVECH##)
LD BC,(DLUG)
LD A,(AUX2##)
AND 9
OR CAO
LD DE,ATARI
LD (CRSR),A
LD A,(DE)
BIT 6,A
JR NZ,ISDRQ
BIT 7,A

JR Z,DRQ?
JR SKONC

ISDRQ: LD A,(HL)
CPL
LD (DR),A
INC HL
DJNZ DRQ?

SKONC: LD A,(AUX2##)
AND 8
JR NZ,NOCRC
LD B,2
DRQ2: LD A,(DE)
AND DRQ
JR Z,DRQ2
LD (DR),A
DJNZ DRQ2
CALL VWDCLR##
JR DOBZE

NOCRC: CALL VWAIIR##

DOBZE: LD A,'C'
RET
ZLE: LD A,'E'
RET

.DEPHASE
END

;NAZWA weakrit
;zapisuje sektor przeznaczony do slabych bitow
;slabe bity zostaja z nagranych wziesniej podkladu,
;poprzez zapisanie sektora tylko okreslona liczba
;bitow, podana,jako parametr

.Z80
ASEG
.RADIX 16
ORG 5000
.PHASE 7800 + 700

INCLUDE DEFS

DLUG EQU $+80
TRACK EQU $+83

CALL VWDCLR##
CALL VMOTGO##
LD A,(TRACK)
LD (TR),A
LD A,(AUX1##)
LD (SR),A
LD HL,(DATVECH##)
LD BC,(DLUG)
LD A,CAO
LD (CRSR),A
DRQ?: LD A,(ATARI)
BIT 6,A
JR Z,DRQ?
LD A,(HL)
CPL
LD (DR),A
INC HL
DJNZ DRQ?

IN A,(WRITOF)
CALL VWAIIR##
IN A,(WRITON)
CALL VWDCLR##
LD A,'C'
RET

.DEPHASE
.REQUEST BAZA
END

;NAZWA ipelad
;sluzy do odczytania grupy pelnych adresow sektorow,
;poczawszy i skonczywszy na otworze indeksowym.
;Reszta jak w pelad. Przeznaczony do odczytywania
;przeplotu synchronizowanych sektorow.

.Z80
ASEG
.RADIX 16
ORG 05000
.PHASE 7800 + 700

INCLUDE DEFS

CZAS EQU $+80

CALL VWDCLR##
CALL VMOTGO##
IN A,(JUMPL0) ;wlacz index fizyczny
LD DE,(CZAS) ;bylo 4000 hex
LD HL,(DATVECH##)
INC HL ;zrob wolne pierwsze miejsce
XOR A ;zeruj licznik
PUSH AF

INDEX?: LD A,(CRSR) ;po clear status ma byc type 1
AND 02 ;czekaj poczatku indexu
JR Z,INDEX? ;ew. potem czekaj konca indexu
ZCZYT: CALL VCZTAD##
DEC HL ;crc nie potrzebujemy
DEC HL
JR Z,TIME
CALL VWAIIR##
AND 08 ;CRC
JR NZ,NOINC
POP AF
INC A
CP 40
JR Z,ZADUZO
PUSH AF
JR ZCZYT

NOINC: DEC HL
DEC HL
DEC HL
DEC HL
JR ZCZYT

TIME: POP AF
ZADUZO: LD HL,(DATVECH##)
LD (HL),A ;wpisz licznik na poczatek bufora
CALL VWDCLR##
LD A,'C'
RET
ZLE: LD A,'E'
RET

.DEPHASE
.REQUEST BAZA
END

;NAZWA iputtrak
;zapisuje sciezke jak puttrack, lecz uzywa otworu
;indeksowego. Przeznaczenie do synchronizowanych sektorow

.Z80
ASEG
.RADIX 16
ORG 05000
.PHASE 7800 + 700

INCLUDE DEFS

LD HL,EXRAM
LD BC,2000
CALL VRECVR##
JP NZ,VDATER##
CALL VACKDA##
CALL VWDCLR##
CALL VMOTGO##
IN A,(JUMPL0) ;wlacz index fizyczny
LD DE,EXRAM+2 ;adres bufora danych
LD HL,ATARI
LD A,OFO ;wbij komende
LD (CRSR),A

JR Z,DRQ?
JR SKONC

ISDRQ: LD A,(HL)
CPL
LD (DR),A
INC HL
DJNZ DRQ?

SKONC: LD A,(AUX2##)
AND 8
JR NZ,NOCRC
LD B,2
DRQ2: LD A,(DE)
AND DRQ
JR Z,DRQ2
LD (DR),A
DJNZ DRQ2
CALL VWDCLR##
JR DOBZE

NOCRC: CALL VWAIIR##

DOBZE: LD A,'C'
RET
ZLE: LD A,'E'
RET

.DEPHASE
.REQUEST BAZA
END

;NAZWA wieloczyt
;wersja a, podaje rozkaz w aux1
;sluzy do odczytu jednego sektora, ktorego numer powtarza
;sie na sciezce, metoda to przed przystapieniem do odczytu
;sektora przeczytac po kolei wszystkie numery sektorow
;poprzezdzajacych, w celu dopasowania sie do przeplotu

.Z80
ASEG
.RADIX 16
ORG 5000
.PHASE 7800 + 700

INCLUDE DEFS

ILSTAR EQU $+0C0 ;ile numerow ozytac na wstepie
TNUM EQU $+0C1 ;numer sciezki
DLUG EQU $+0C2 ;dlugosc sektora
LISTAR EQU $+0C4 ;lista numerow sektorow

CALL VWDCLR##
CALL VMOTGO##
LD HL,VCZYT##+1
LD C,(HL)
INC HL
LD B,(HL)
INC BC
INC BC
LD (JUMP+1),BC
LD C,30
AGA: LD HL,LISTAR
CALL ADR
JR NZ,AGA
LD A,(ILSTAR)
LD B,A
INC HL
CIAG: CALL ADR
JR Z,WCIAGU

DOKONC: CALL ADR
DJNZ DOKONC
DEC C
JR NZ,AGA
JR ZLE

WCIAGU: INC HL
DJNZ CIAG

GETIT: LD A,(TNUM)
LD (TR),A
LD A,(HL)
LD (SR),A
LD HL,(DATVECH##)
LD BC,(DLUG)
LD A,(AUX1##)
CALL JUMP
JR Z,ZLE
CALL VWAIIR##
LD (WDSTAT##),A
LD A,'C'
RET
ZLE: LD A,'E'
RET

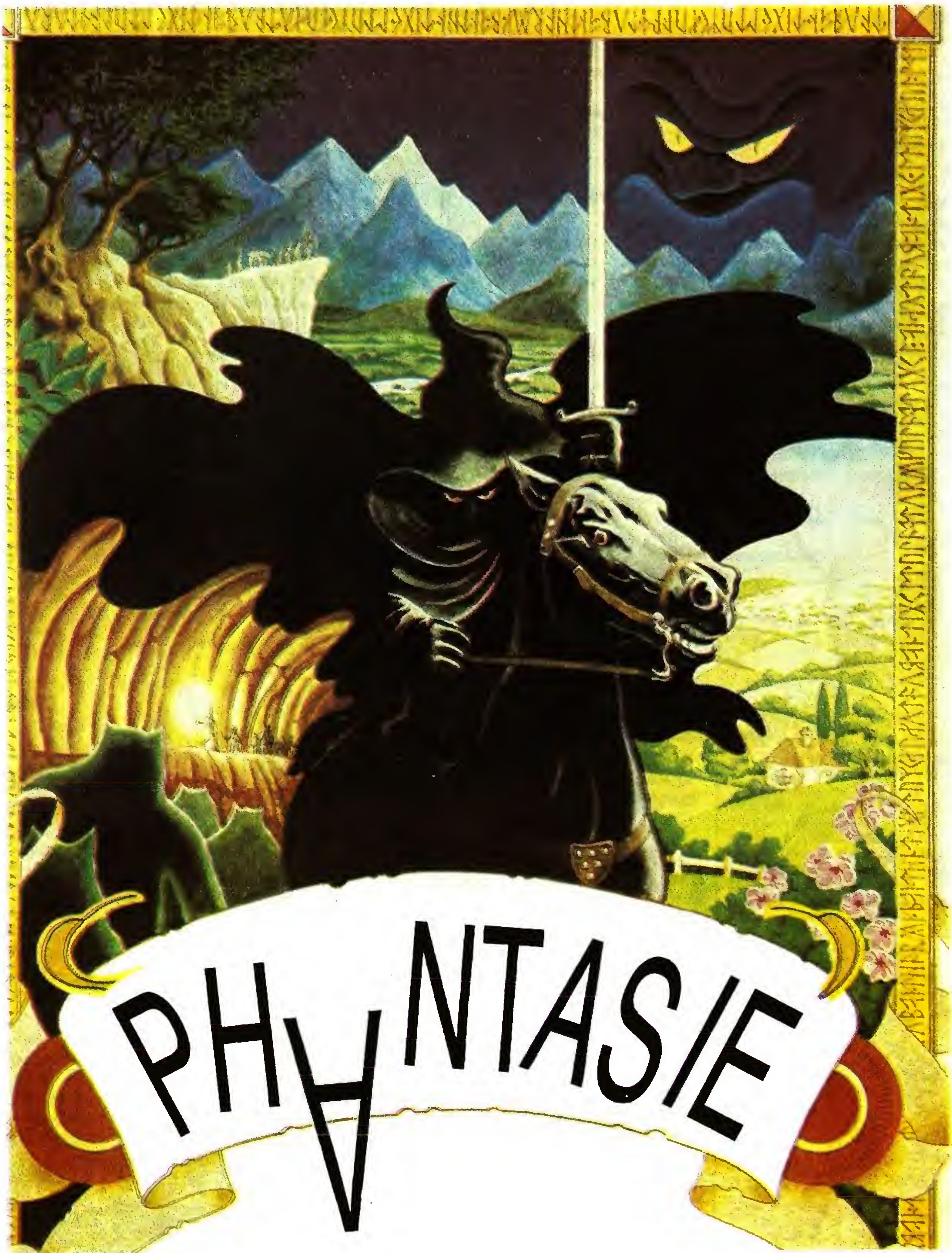
JUMP: JP 1111
ADR: PUSH BC
PUSH HL
AD1: CALL VREAAD##
JR NZ,NIEZLE
POP HL
POP HL
POP HL
JR ZLE

NIEZLE: CALL VWAIIR##
AND 08
JR Z,NOCRC
CALL VWDCLR##
JR AD1
POP HL
POP BC
LD A,(SEC##)
CP (HL)
RET

.DEPHASE
.REQUEST BAZA
END

***** KONIEC LISTINGOW ROZSZERZEN *****

```



„Phantasie” jest przygodową grą fantastyczną typu *roleplaying*. Zadaniem gracza jest skompletowanie drużyny, a następnie wraz z nią uwolnienie wyspy Gelnor od zamieszkujących ją potworów.

LUDY

Rasy

- HUMAN – ludzie
- DWARF – krasnoludy
- ELF – elfy
- GNOM – gnomy
- HALFLINGE – półelfy
- GNOLL – gnolle
- GOBLIN – gobliny
- LIZZ. MAN – ludzie-jaszczury
- COBOLD – koboldy
- MINOTAUR – półludzie-półbyki
- OGRE – wilkołaki
- ORK – orki
- PIXIE – chochliki
- SPRITES – skrzaty
- TROLL – trole

Zawody

- FIGHTER – wojownik
- MONK – mnich
- PRIEST – kapłan
- RANGER – strażnik
- THIEF – włamywacz
- WIZARD – czarodziej

Cechy

- STRENGHT – siła
- INTELLIGENCE – inteligencja
- DEXTERITY – zręczność
- CONSTITUTION – kondycja
- CHARISMA – charyzma
- LUCK – szczęście

MIASTA

- GUILD (gildia) – skład i nauka drużyny

- ARMORY (zbrojownia) – zakup wyposażenia
- BANK (bank) – operacje finansowe
- INN (hotel) – wypoczynek, rozdzielanie wyposażenia
- MYSTIC (pogłoski) – wskazówki
- EXIT (wyjście) – opuszczenie miasta
- CAST SPELL – użycie zaklęcia
- USE ITEM – użycie przedmiotu
- GAME SAVE – zapisanie gry
- PRINTOUT – wydruk

Gildia

- SPELLS – nauka nowych zaklęć (z listy)
- TRAINING – trening postaci (podnoszenie poziomu)
- ADD – dołączenie postaci do drużyny
- NEW – tworzenie nowej postaci
- DROP – usunięcie postaci z drużyny

RENAME – zmiana imienia postaci
 INSPECT – sprawdzenie danych postaci
 AGE – wiek
 HIT POINTS – punkty trafienia (życia)
 MAGIC POINTS – punkty magii
 GOLD IN BANK – złoto w banku
 EXPERIENCE – doświadczenie
 SKILLS – umiejętności
 ATTACK – atak
 PARRY – obrona
 SWIM – pływanie
 LISTEN – słuch
 SPOT TRAP – usuwanie pułapek
 DISARM – rozbrajanie
 FIND ITEMS – znajdowanie skarbów
 PICK LOCK – otwieranie zamków
 EQUIPMENT – wyposażenie: broń, zbroja i tarcza
 SPELL – lista opanowanych zaklęć
 PURGE – skasowanie postaci
 EXIT – wyjście z gildii

CZARY

Nr	Koszt (MP)	Nazwa	Rodzaj	Działanie
1	1	HEAL 1	W	uzdrowienie + 1-9 HP
2	2	HEAL 2	W	uzdrowienie + 1-27 HP
3	3	HEAL 3	W	uzdrowienie + 1-57 HP
4	4	HEAL 4	W	uzdrowienie + 1-99 HP
5	1	FIRE 1	B/A	kula ognista — 1-10 HP
6	2	FIRE 2	B/A	kula ognista — 1-30 HP
7	3	FIRE 3	B/A	kula ognista — 1-60 HP
8	4	FIRE 4	B/A	kula ognista — 1-100 HP
9	1	SPEED 1	B/P	skuteczność + 20%
10	2	SPEED 2	B/P	skuteczność + 25%
11	3	SPEED 3	B/P	skuteczność + 45%
12	4	SPEED 4	B/P	skuteczność + 70%
13	1	FORCE 1	B/P	siła + 1-2 ran
14	2	FORCE 2	B/P	siła + 1-4 ran
15	3	FORCE 3	B/P	siła + 1-7 ran
16	4	FORCE 4	B/P	siła + 1-11 ran
17	1	PARRY 1	B/P	odporność tarczy + 1
18	2	PARRY 2	B/P	odporność tarczy + 2
19	3	PARRY 3	B/P	odporność tarczy + 3
20	4	PARRY 4	B/P	odporność tarczy + 4
21	1	FUSS 1	B/A	sieje zamęt u wroga
22	2	FUSS 2	B/A	sieje zamęt u wroga
23	3	FUSS 3	B/A	sieje zamęt u wroga
24	4	FUSS 4	B/A	sieje zamęt u wroga
25	1	WEAK 1	B/A	osłabia ciosy wroga
26	2	WEAK 2	B/A	osłabia ciosy wroga
27	3	WEAK 3	B/A	osłabia ciosy wroga
28	4	WEAK 4	B/A	osłabia ciosy wroga
29	1	BIND 1	B/A	krępuje ruchy wroga
30	2	BIND 2	B/A	krępuje ruchy wroga
31	3	BIND 3	B/A	krępuje ruchy wroga
32	4	BIND 4	B/A	krępuje ruchy wroga
33	1	PSYCH.SHOCK 1	B/A	szok psych. — 1-5 HP
34	2	PSYCH.SHOCK 2	B/A	szok psych. — 1-11 HP
35	3	PSYCH.SHOCK 3	B/A	szok psych. — 1-31 HP
36	4	PSYCH.SHOCK 4	B/A	szok psych. — 1-66 HP
37	1	FIRE FLAME 1	B/A	plomień — 1-10 HP
38	2	FIRE FLAME 2	B/A	plomień — 1-30 HP
39	3	FIRE FLAME 3	B/A	plomień — 1-60 HP
40	4	FIRE FLAME 4	B/A	plomień — 1-100 HP
41	1	CHARME	B/A	wróg zaprzestaje walki
42	2	SLEEP	B/A	wróg usypa
43	3	TELEPORT	B/A	teleportacja drużyny
44	4	ENLIVEN	W	przywrócenie życia
45	1	NINJA 2	B/P	przekształcenie w ninję
46	2	FEAR	B/A	panika wśród monstrów
47	3	NOTHING	B/A	zniknięcie wroga
48	4	CALL ELEM.	B/A	przywołanie żywiołu
49	1	ZOMBIES	B/A	atakuję zombie
50	2	NINJA 1	B/P	przekształcenie w ninję
51	3	AWAKEN	B/P	obudzenie drużyny
52	4	VALUE	N/PJ	poznanie siły wroga
53	1	VISION	N/P	dalekie widzenie
54	2	TRANSPORT	N/M	ruch drużyny

Rodzaje czarów:

B – zaklęcia bojowe (tylko w walce)
 B/A – niszczą wroga
 B/P – przeciwdziałają czarom
 N – zaklęcia normalne
 N/M – tylko w miastach
 N/P – tylko w otwartej przestrzeni
 N/J – tylko w jaskiniach
 W – zaklęcia wszechstronne

Bank

WITHDRAW – podejmowanie zdeponowanego złota
 DEPOSITE – wpłacanie złota do banku
 EXIT – wyjście z banku

Zbrojownia

BUY – zakup wyposażenia
 NEXT SHOPPER – następny nabywca
 EXIT – wyjście ze zbrojowni

WYPRAWA

MOVE – dalszy ruch
 CAST – użycie zaklęcia
 HEALTH – raport o zdrowiu drużyny
 USE – użycie eliksiru

Lochy

LOOK – poszerzenie widoku
 LISTEN – nasłuchiwanie
 PICK LOCK – otwarcie zamka
 FIND ITEM – poszukiwanie skarbów
 AVOID TRAP – usuwanie pułapek

Walka

BEG MERCY – błaganie o łaskę
 FIGHT – walka z wrogiem
 THRUST – lekki atak
 ATTACK – normalny atak
 SLASH – silny atak
 LUNGE – słaby, wielokrotny atak
 SPELL – rzucanie zaklęć
 PARRY – obrona przed ciosami
 PARTY IS SLOWED – drużyna straciła szybkość
 PARTY IS WEAKED – drużyna jest osłabiona
 STUNNING BLOW – postać została uśpioną
 MAGIC IS DRAINED – wyczerpanie magii
 THREATEN – straszenie wrogów
 GREETING – powitanie wrogów
 FLEE – wycofanie
 RUN – ucieczka
 SURRENDER – poddanie się

WYPOSAŻENIE

Wymagana siła = 2* STRENGTH + DEXTERITY

Tarcze

Nr	Wartość	Wymagana siła	Nazwa
1	1	1	GLOVES
2	2	2	WOOD SHIELD
3	3	3	WOOD SHIELD
4	4	4	SMALL SHIELD
5	5	5	SMALL SHIELD +1
6	6	6	SMALL SHIELD +2
7	7	7	SMALL SHIELD +3
8	8	8	MEDIUM SHIELD
9	9	9	MEDIUM SHIELD +1
10	10	10	MEDIUM SHIELD +2
11	11	11	MEDIUM SHIELD +3
12	12	12	LARGE SHIELD
13	13	13	LARGE SHIELD +1
14	14	14	LARGE SHIELD +2
15	15	15	LARGE SHIELD +3
16	16	16	GIANT SHIELD
17	17	17	GIANT SHIELD +1
18	18	18	GIANT SHIELD +2
19	19	19	GIANT SHIELD +3
20	20	20	GODS SHIELD

Zbroje

Nr	Wartość	Wymagana siła	Nazwa
21	1	2	CLOTHING
22	2	4	ROBE
23	3	6	LEATHER
24	4	8	HEAVY LEATHER
25	5	10	RING MAIL
26	6	12	SCALE MAIL
27	7	14	CHAIN MAIL
28	8	16	SPLINT MAIL
29	9	18	BANDED MAIL
30	10	20	PLATE MAIL
31	2	0	CLOTHING +1
32	3	0	ROBE +1
33	4	2	LEATHER +1
34	5	4	LEATHER +2
35	6	6	RING MAIL +1
36	7	8	RING MAIL +2
37	8	10	CHAIN MAIL +1
38	9	12	CHAIN MAIL +2
39	10	14	GODS CLOTHING
40	11	16	GODS ARMOUR

Broń

Nr	Wartość	Wymagana siła	Nazwa
41	0	2	STICK
42	1	4	KNIFE
43	1	6	SMALL CLUB
44	2	8	SMALL STAFF
45	2	10	SMALL MACE
46	3	12	DAGGER
47	3	14	SMALL FLAIL
48	4	16	CLUB
49	4	18	MACE
50	5	20	SMALL HAMMER
51	5	22	SMALL AXE
52	6	24	STAFF
53	6	26	SHORT SWORD
54	7	28	FLAIL
55	7	30	HAMMER
56	8	32	PITCH FORK
57	8	34	SPEAR
58	9	36	AXE
59	9	38	SWORD
60	10	40	LARGE MACE
61	10	42	MAUL
62	11	44	TRIDENT
63	11	46	LARGE SPEAR
64	12	48	LARGE AXE
65	12	50	MOORNING STAR
66	13	52	PIKE
67	13	54	LONG SWORD

68	14	56	SPADE
69	14	58	SETUM
70	15	60	HELBED
71	3	2	SMALL MACE +1
72	4	4	DAGGER +1
73	4	6	SMALL MACE +2
74	5	8	DAGGER +2
75	6	10	DAGGER +3
76	7	12	STAFF +1
77	7	14	DAGGER +4
78	8	16	FLAIL +1
79	9	18	SPEAR +1
80	10	20	AXE +1
81	10	22	SWORD +1
82	11	24	SWORD +2
83	12	26	SWORD +3
84	13	28	LARGE AXE +1
85	13	30	SWORD +4
86	14	32	SWORD +5
87	15	34	SWORD +6
88	16	36	HELBED +1
89	16	38	SWORD +7
90	17	40	HELBED +2
91	18	42	HELBED +3
92	19	44	SWORD +10
93	19	46	HELBED +4
94	20	48	HELBED +5
95	21	50	HELBED +6
96	22	52	HELBED +7
97	22	54	GODS KNIFE
98	23	56	GODS MACE
99	24	58	GODS AXE
100	25	60	GODS SWORD

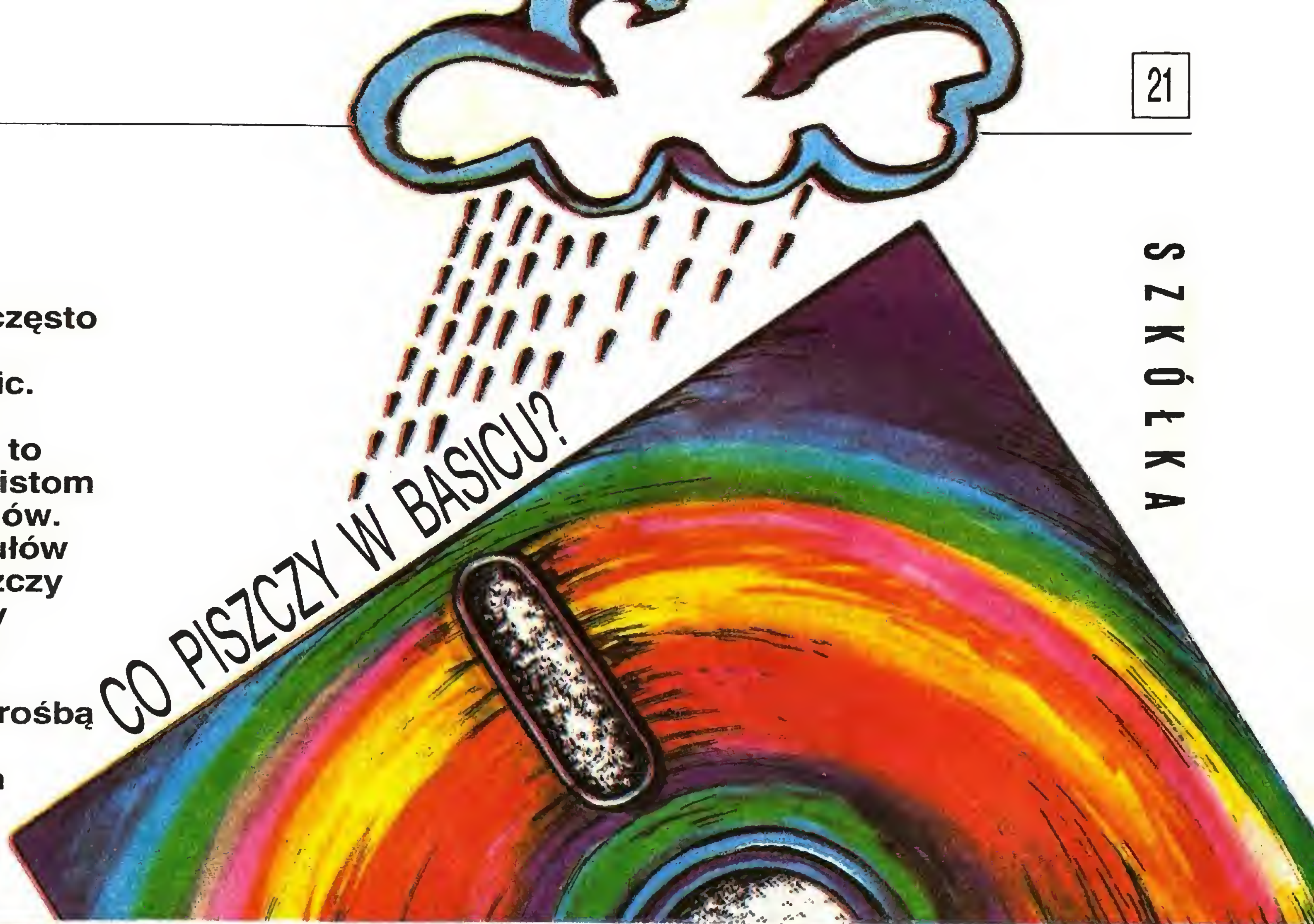
Eliksiry

Nr	Nazwa	Odnowienie
101	HEAL POTION 1	1 HP
102	HEAL POTION 2	4 HP
103	HEAL POTION 3	9 HP
104	HEAL POTION 4	16 HP
105	HEAL POTION 5	25 HP
106	HEAL POTION 6	36 HP
107	HEAL POTION 7	49 HP
108	HEAL POTION 8	64 HP
109	HEAL POTION 9	81 HP
110	HEAL POTION 10	100 HP
111	MAGIC POTION 1	3 MP
112	MAGIC POTION 2	6 MP
113	MAGIC POTION 3	9 MP
114	MAGIC POTION 4	12 MP
115	MAGIC POTION 5	15 MP
116	MAGIC POTION 6	18 MP
117	MAGIC POTION 7	21 MP
118	MAGIC POTION 8	24 MP
119	MAGIC POTION 8	27 MP
120	MAGIC POTION 10	30 MP



Nasi Czytelnicy w listach i telefonach do redakcji pytają często o różne zagadnienia związane z programowaniem w Atari Basic. Mimo, iż język ten jest stary i — zdawałoby się — dobrze znany, to jednak początkującym programistom sprawia on nadal wiele problemów. Rozpoczynamy więc cykl artykułów pod wspólnym tytułem „Co piszczy w Basicu?”, w którym będziemy opisywać sposoby uzyskiwania różnych efektów w Atari Basic. Jednocześnie zwracamy się z prośbą do Czytelników o nadsyłanie propozycji zagadnień, o których chcieliby przeczytać.

CO PISZCZY W BASICU?



WPROWADZANIE

Jednym z najczęściej napotykanych problemów jest wprowadzanie danych do programu. Szczególnie chodzi tu o przekazywanie programowi prostych decyzji użytkownika w rodzaju odpowiedzi na pytanie: „tak czy nie”. Większość początkujących programistów wykorzystuje w takich przypadkach instrukcję INPUT.

Instrukcja INPUT jest w zasadzie przeznaczona do wprowadzania danych dłuższych niż jeden znak. Z tego powodu wymaga zasygnalizowania, że zakończono już wprowadzanie informacji, naciśnięciem klawisza <RETURN>. Jeżeli program żąda wyboru wariantu, to użytkownik zamiast jednego klawisza musi nacisnąć dwa. Na przykład:

```
1000 ? "CHCESZ INSTRUKCJE (T/N) ?";
1010 INPUT T$
1020 IF T$="T" THEN GOSUB 2000
1030 IF T$<>"N" THEN 1010
```

Każda odpowiedź zarówno „tak”, jak i „nie” wymaga w takim przypadku naciśnięcia kolejno klawiszy <T> lub <N> i <RETURN>. Usuwając z tego przykładu wiersz 1030 upraszczamy nieco obsługę programu, gdyż zamiast odpowiedzi „nie” wystarczy tylko nacisnąć <RETURN>. Rezultatem będzie bowiem ciąg pusty („”), który jest różny od ciągu „T” i nie spowoduje wywołania procedury od wiersza 2000. Jednak użycie tej konstrukcji programowej dla wybrania jednego z kilku wariantów zmusza za każdym razem do naciskania dwóch klawiszy. Na przykład:

```
1000 ? "WYBIERZ WARIANT"
1010 ? "(0) - ODCZYT PLIKU"
1020 ? "(Z) - ZAPIS PLIKU"
1030 ? "(K) - KONIEC PROGRAMU"
1040 INPUT T$
1050 IF T$="0" THEN 2000
1060 IF T$="Z" THEN 3000
1070 IF T$<>"K" THEN 1040
1080 END
```

Przy okazji warto zwrócić uwagę na sposób sprawdzania, czy użytkownik udzielił poprawnej odpowiedzi. Większość początkujących programistów stojąc przed zagadnieniem pokazanym w ostatnim przykładzie stosuje konstrukcję:

```
1070 IF T$="K" THEN 1090
1080 GOTO 1040
1090 END
```

Istnieje jednak w Atari Basic instrukcja, która pozwala na prostsze rozwiązanie tego problemu. Jest to instrukcja GET. Pobiera ona jeden bajt danych ze wskazanego kanału wejścia/wyjścia, który oczywiście musi być uprzednio otwarty. Po zastosowaniu tej instrukcji nasz pierwszy przykład przybierze postać:

```
1000 ? "CHCESZ INSTRUKCJE (T/N) ?";
1010 OPEN #3,4,0,"K":GET #3,T:CLOSE #3
1020 IF T=84 THEN GOSUB 2000
1030 IF T>78 THEN 1010
```

Teraz wystarczy naciśnięcie jednego klawisza, aby komputer zrozumiał, o co chodzi użytkownikowi. Je-

śli takich pytań jest w programie więcej, to kanał odczytu klawiatury można otworzyć jednorazowo na początku programu. W takim razie nie trzeba go już zamykać — będzie on zamknięty automatycznie po wykonaniu ostatniego wiersza programu, instrukcji END, RUN, LOAD lub CLOAD. Jeżeli programista nie zna kodów ASCII odpowiednich liter (choć powinien), to może sobie ułatwić pracę pisząc:

```
1020 IF T=ASC("T") THEN GOSUB 2000
1030 IF T<>ASC("N") THEN 1010
```

Przy odczytywaniu znaku z klawiatury należy jeszcze pamiętać o możliwości przełączenia jej na pisanie małymi literami. W takim wypadku nawet naciśnięcie <T> spowoduje wybranie wariantu „nie” (również przy zastosowaniu INPUT). Dla komputera nie ma bowiem żadnego związku pomiędzy znakami „T” i „t”. Aby uniknąć takiej sytuacji, trzeba wprowadzić do naszego przykładu także sprawdzanie drugiej możliwości:

```
1020 IF T=84 OR T=116 THEN GOSUB 2000
1030 IF T<>78 AND T<>110 THEN 1010
```

Użycie GET jest szczególnie przydatne przy wybieraniu z menu wielu wariantów działania programu, jeśli warunki te oznaczmy kolejnymi liczbami. Wartości zwracane przez GET będą w takim wypadku kolejnymi liczbami, co pozwala na wykorzystanie wielokrotnej instrukcji warunkowej ON (większość programistów zapomina całkowicie o jej istnieniu). Instrukcja ON wybiera bowiem wariant w zależności od wartości umieszczonego w niej wyrażenia warunkowego. Można to zastosować następująco:

```
1000 ? "WYBIERZ WARIANT"
1010 ? "(1) - ODCZYT PLIKU"
1020 ? "(2) - ZAPIS PLIKU"
1030 ? "(3) - SKASOWANIE PLIKU"
1040 ? "(4) - ZMIANA NAZWY PLIKU"
1050 ? "(0) - KONIEC PROGRAMU"
1060 GET #3,T
1070 ON T-47 GOTO 1090,2000,3000,4000,5000
1080 GOTO 1060
1090 END
```

System operacyjny Atari daje jeszcze jedną możliwość rozwiązania zagadnienia odczytu z klawiatury pojedynczego znaku. Wykorzystuje się w tym celu rejestr KBCODE (adres 764), w którym przechowywany jest kod ostatnio naciśniętego klawisza. Niestety nie jest to kod ASCII, lecz specjalny wewnętrzny kod Atari. Zanim wykorzystamy ten rejestr do odczytu znaku, najpierw trzeba powiedzieć kilka słów o jego działaniu.

Jeżeli użytkownik nie nacisnął żadnego klawisza, to w rejestrze KBCODE znajduje się wartość 255. Naciśnięcie dowolnego klawisza (z wyjątkiem <SHIFT>, <CONTROL> i <BREAK>) powoduje wpisanie do tego rejestru wartości kodu tego klawisza. Kod ten może być zmodyfikowany przez dodanie 64, jeśli równocześnie był wciśnięty <SHIFT>, 128, jeśli był wciśnięty <CONTROL> lub 192, jeśli wciśnięte były oba. Kod ten pozostaje w rejestrze tak długo, dopóki

nie zostanie odczytany przez Basic instrukcją INPUT lub GET. Przy korzystaniu z KBCODE trzeba więc pamiętać, aby skasować jego zawartość po wykorzystaniu. W przeciwnym razie pozostający tam kod może powodować błędy w innych częściach programu.

Przy wykorzystaniu bezpośredniego odczytu z KBCODE można nasz pierwszy przykład przekształcić następująco:

```
1000 ? "CHCESZ INSTRUKCJE (T/N) ?";
1010 POKE 764,255
1020 IF PEEK(764)=45 THEN GOSUB 2000
1030 IF PEEK(764)<>35 THEN 1020
1040 POKE 764,255
```

Wartość odczytaną z rejestru klawiatury można oczywiście przypisać dowolnej zmiennej, a następnie poddawać ją dalszej obróbce. W ten sposób łatwo można wyeliminować wpływ klawiszy <SHIFT> i <CONTROL> oraz ustawienia klawiatury na pisanie małymi lub dużymi literami. Ma to większe znaczenie przy wybieraniu z rozbudowanego menu, lecz pokażemy to na naszym krótkim przykładzie:

```
1000 ? "CHCESZ INSTRUKCJE (T/N) ?";
1010 POKE 764,255
1020 IF PEEK(764)=255 THEN 1020
1030 T=PEEK(764):POKE 764,255
1040 IF T>192 THEN T=T-192
1050 IF T>128 THEN T=T-128
1060 IF T>64 THEN T=T-64
1070 IF T=45 THEN GOSUB 2000
1080 IF T<>35 THEN 1020
```

Teraz program zareaguje prawidłowo nawet wtedy, gdy użytkownik zamiast klawisza <T> naciśnie kombinację <SHIFT><CONTROL><T>.

Wykorzystanie rejestru KBCODE umożliwia dodatkowo zasymulowanie instrukcji INKEY\$, której Atari Basic nie posiada. Wystarczy tylko uzależnić odczytywanie znaku od naciśnięcia klawisza i to wszystko:

```
1000 IF PEEK(764)<>255 THEN GET #3,T
```

A jeśli chcemy zasymulować instrukcję INKEY\$, to należy jeszcze dodać zamianę kodu ASCII na znak:

```
1000 IF PEEK(764)<>255 THEN GET #3,T:
T$=CHR$(T)
```

Oczywiście kanał I/O numer 3 musi być uprzednio otwarty do odczytu z klawiatury. Zauważ, że nie trzeba po odczytaniu znaku wpisywać do KBCODE wartości 255, gdyż instrukcja GET robi to automatycznie.

Czy zatem instrukcja INPUT jest całkowicie zbędna? Ależ nie! Należy ją jednak wykorzystywać zgodnie z przeznaczeniem, czyli do wprowadzania danych o długości większej niż jeden znak. Ale o tym napiszę w następnym numerze „Mojego Atari”.

Wojciech ZIENTARA

D
A
N
Y
C
H

XL/XE

BLINKY'S SCARY SCHOOL Zeppelin Games przygodowa	C/D	Gra jest bardzo ładnie opracowana, ciekawa fabuła, wymaga nie tylko zręczności, ale także logicznego myślenia. Nic więc dziwnego, że przygody sympatycznego duszka w starym zamczysku goszczą po raz trzeci na naszej liście.
THE GOONIES Activision przygodowa	C/D	Program opracowano kilka lat temu, ale nadal posiada bardzo dużo zalet. Grafika bez zastrzeżeń, nie przeszkadzająca muzyka, a do tego znakomita opcja gry dla dwóch graczy. Zagadki do rozwiązania nie takie znów trudne, więc nic dziwnego, że również po raz trzeci na naszej liście.
INFILTRATOR U.S. Gold przygodowa	D	Takich gier brakuje na „małe” ATARI. Misja specjalna, składająca się z kilku etapów, w jednej grze kilka zupełnie różnych części: lot helikopterem, akcja w bazie itp. Niestety sprzedawana na kasetach pod tym tytułem tylko niewielka część gry, praktycznie nie do zabawy.
BOULDER DASH First Star Software zręcznościowa	C/D	Gra, która jest prawdziwym hitem, a jej wykonanie na „małe” ATARI znakomite. Wędrujący i zbierający diamenty Rockford doczekał się już ponad 10 wersji programu (oficjalnie tylko 4). W grze oprócz zręczności należy wykazać się umiejętnością szybkiego, logicznego myślenia i kojarzenia, co jest jej największym atutem.
M.U.L.E. Electronic Arts ekonomiczna	C/D	Znakomita gra dla kilku osób. Na nieznanym planecie ląduje statek z uczestnikami gry, każdy dostaje działkę i musi sobie radzić, aż do następnego przybycia statku: buduje się działki, uprawia ziemię, handluje, kupuje. Gra, w którą można grać bardzo długo i się nie znudzi, polecana jednak głównie starszym użytkownikom komputerów.
TIGER ATTACK Atari Corp. zręcznościowa	C/D	Gra do złudzenia przypomina znany z „automatów” hit — „Flying Shark”. Niewielki samolotek nad dżunglą i atakujący go przeciwnicy, szybka strzelanina w ładnej grafice, bardzo dobra muzyka. Gra typowo zręcznościowa, jednak bardzo wciągająca.
F-15 STRIKE EAGLE Microprose symulacyjna	C/D	Autorem programu jest Sid Meier, jeden z najlepszych specjalistów od gier symulacyjnych. Akcja programu rozgrywa się wyłącznie w powietrzu, ale mnogość opcji powoduje, że nawet bez elementów startu, czy lądowania jest ona bardzo atrakcyjna. Kilka misji, cele naziemne i powietrzne do zniszczenia...
KARATEKA Broderbound Software przygodowa	C/D	Jeden z niewielu programów „karate”, wyróżnia go jednak od pozostałych fantastyczna wręcz animacja, majestatycznie poruszająca się duża postać bohatera, sprytni i szybcy przeciwnicy oraz piękna księżniczka Marika do uwolnienia, świetna muzyka — jednym słowem perfekcyjnie opracowana gra.
BLACK LAMP Atari Corp. przygodowa	C/D	Sympatyczny błazen wędruje po kolorowym pałacu, ma wielu wrogów i przeciwników, co krok „spotyka” niespodzianki. Bardzo ładnie i starannie opracowana gra, zresztą znana także z innych komputerów. Jest to jedna z nowych gier, które firma ATARI wypuściła na rynek europejski w celu promocji komputerów serii XL/XE.
MR DO! Datasoft zręcznościowa	C/D	Bardzo szybka, ale i bardzo atrakcyjna, szczególnie gdy gra kilka osób. Pracowity pan Do zbiera w swoim sadzie owoce, przeszkadzają mu w tym całe hordy insektów. Jedyną jego bronią jest szybkość i kulka na uwięzi. Kilkanaście poziomów i dopingująca do „walki” tablica najlepszych wyników podnoszą atrakcyjność gry.
WORLD SOCCER Zeppelin Games ekonomiczna	C/D	Gra reprezentująca jeden z najbardziej modnych rodzajów programów. Zostałeś managerem drużyny piłkarskiej, wybierasz skład, ustalasz taktykę na poszczególne mecze, kierujesz treningami. Wzbogacona o ciekawą grafikę i sprawozdania z meczy, wydaje się być godną polecenia.

ST

KICK OFF II — FINAL WHISTLE Anco sportowa	Firma Anco wypuściła na rynek już chyba ostatni program z serii Kick Off, sugeruje to podtytuł „Końcowy gwizdek”. Najnowsza wersja piłki nożnej w stosunku do poprzednich została wzbogacona o kilka nowych opcji i tak jak one nie pozwala oderwać się od komputera.
DIZZY — TREASURE ISLAND Code Master przygodowa	Wspaniała grafika, prześmieszna posać bohatera-jajeczka i oczywiście wciągająca fabuła decydują o wartości tej gry. Dizzy wędruje po kilku wyspach znajdując różne przedmioty, które przydają mu się w dalszej wędrówce, aż po zdobycie upragnionego skarbu. Podstawową wartością gry jest także to, że oprócz zręczności należy się wykazać zdolnością logicznego myślenia i kojarzenia.
OIL IMPERIUM Reline Software ekonomiczna	Program spełnia marzenia wielu z nas — pieniądze, wielki biznes, luksusy, a wszystko za sprawą... ropy naftowej. Zostałeś potentatem naftowym, aby osiągnąć sukces musisz jak najlepiej poprowadzić swoje interesy, rozbudowując sieć wydobywczą i przetwórczą. Gra znakomita szczególnie dla kilku osób.
LEMMINGS Psygnosis logiczna	Rewelacja roku! Gra zupełnie nowego rodzaju. Oczywiście grafika i dźwięk znakomite. Zadaniem gracza jest pokierowanie grupą śmiesznych stworzonek zwanych Lemmingami. Każdy z nich posiada różne zdolności, które należy odpowiednio wykorzystać, aby wyprowadzić pozostałych z jaskiń, gra z dnia na dzień przebojem zdobywa rynki na całym świecie.
SHERMAN M-4 Loricel symulacyjna	Tym razem program symulujący prowadzenie czołgu. Gracz uczestniczy w jednej z wybranych kampanii wojennych i posiada określone zadania do wykonania. Firma Loricel gwarantuje — jak zawsze — znakomitą grafikę i wiele bardzo ciekawych opcji, a ponieważ prowadzenie czołgu nie jest bardzo trudne, więc gra może się podobać „i młodym i starym”.
PRO TENNIS TOUR II — GREAT COURTS II UBI Soft. sportowa	Najnowszy program tenisowy — rewelacja przede wszystkim ze względu na kilka poziomów gry, od łatwego, gdzie można po krótkim treningu wygrać nawet z komputerem, po bardzo trudny, gdzie mamy problemy z odbiciem piłki. Oczywiście wiele ciekawych opcji: udział w cyklu turniejów Grand Prix, gra pojedyncza, podwójna, różne rodzaje nawierzchni, programowanie treningu z maszyną, itp.
GODS Renegade zręcznościowa	Grafika i dźwięk są bliskie ideału, świetna zabawa we wspaniałych kolorach. Bohaterem jest Hercules, który wędrując po niebezpiecznej krainie musi pokonać cztery poziomy strzeżone przez strażników, pułapki i demony, aby dotrzeć do legendarnego miasta bogów, gdzie czeka go ostateczne i najniebezpieczniejsze spotkanie.
TEST DRIVE II — THE DUEL Accolade auto-moto	Nowa wersja znakomitego programu symulacyjnego jazdy samochodem. W stosunku do pierwszej wersji rozszerzono znacznie zakres programu, pojawiły się nowe opcje związane m.in. ze stopniem trudności, a przede wszystkim możliwość wczytywania nowych tras i nowych samochodów.
METAL MUTANT Palace zręcznościowa	W odległym świecie, w odległej przyszłości mutanty robotów walczą o życie. Znakomita grafika, do perfekcji dopracowano nawet szczegóły tła, świetnie animowane postacie wojowników, bardzo dobra muzyka, a odgłosy walki i „przyrody” bliskie ideału. Bardzo cenna opcja zapisu stanu gry.
9 LIVES ARC Software labiryntowa	Powszechnie wiadomo, że każdy kot ma „9 żyć”, więc tytułowy bohater jest sympatycznym kotem, który wędruje po świecie pełnym stworków, duszków i różnych żyjątek. Kot jest świetnie animowany, robi śmieszne miny i potrafi sobie znakomicie radzić; idąc wśród pułapek i niebezpieczeństw zbiera cenne przedmioty.
JAMES POND Millenium zręcznościowa	Tytuł sugeruje parodię filmu z agentem 007, James Pond to podwodny super agent M15 — śmieszna rybka, która pływając przez 12 poziomów musi oczyścić królewskie wody z niedobrych stworków. Gra posiada przyjemną dla oka grafikę, a poza tym śmieszy i bawi.

Na zakończenie kilka słów na temat nowości:

LAST NINJA III (System 3) — najnowszy program z tej serii, wyróżnia się jeszcze lepszą grafiką. Fabuła i zasada gry podobna jak w poprzednich. Duże, ładnie animowane postacie i kilkunastu przeciwników na drodze do upragnionego celu, znakomite zakończenie gry.

MIDWINTER II — FLAMES OF FREEDOM (Rainbird) — akcja nowej wersji znakomitego programu przenosi się z krainy lodów na zielono-niebieskie słoneczne wyspy. Twoim zadaniem jest uratowanie pokoju dla świata.

DRAGON'S LAIR II — TIMEWARP (Empire) — niedobry czarownik Mordroc uwięził piękną księżniczkę Daphne, tylko Dirk the Daring może ją uwolnić. Kolejny program z serii znakomitych animowanych komiksów.

Tomasz MAZUR

GATO

XL/XE

„Gato” jest programem symulującym pływanie okrętem podwodnym. W symulacji tej dowodzi się amerykańskim okrętem podwodnym typu „Gato”, który jest wyposażony w radar oraz liczne torpedy i miny. Podobnie jak w innych grach tego rodzaju można samodzielnie regulować prędkość na- i podwodną (maksymalna prędkość wynosi 20 węzłów).

W przeciwieństwie do klasycznego „Silent Service” gra zawiera różnorodne zadania, które należy kolejno realizować. Są to m.in. ataki na konwoje statków i akcje ratowania zestrzelonych lotników, przy czym okręt jest atakowany przez niszczyciele i eskortowce. Gracz powinien wykonać możliwie jak najwięcej zadań bez powracania do bazy w celu załadowania torped lub wykonania napraw.

Po bardzo ładnie wykonanej planszy tytułowej ukazuje się menu, z którego można wybrać różne warianty gry. Są tu m.in. stopień trudności, liczba graczy, pora doby, włączenie lub

wyłączenie dźwięku. Po rozpoczęciu gry znajdujemy się na pomoście okrętu podwodnego. W dolnej połowie ekranu widoczne są przyrządy wskazujące zanurzenie, prędkość, kurs, zapas paliwa, stan naładowania akumulatorów, liczbę torped, rodzaj napędu oraz odniesione uszkodzenia. Teraz otrzymujemy zadanie do wykonania. Na życzenie gracza może ono być przekazane alfabetem Morse’a.

„Gato” — podobnie jak „Silent Service” — jest obsługiwane również z klawiatury. Po przez naciśnięcie klawisza można, na przykład, wywołać mapę, na której przedstawiony jest rejon akcji podzielony na kwadraty. Bardzo ważne dla przetrwania jest korzystanie z radaru, który informuje o okrętach przeciwnika, wyspach i niebezpiecznych rafach. Te ostatnie można napotkać przy wszystkich wyspach. Ponadto można wywołać dziennik okrętowy lub zaryzykować spojrzenie przez peryskop. Aktualny stan gry może być zapisany na dyskietkę, co warto

zrobić przed szczególnie niebezpiecznymi zadaniami.

Wszystkie statki i okręty są na ekranie wyświetlane wektorowo (co jednak nie pozwala na rozróżnienie szczegółów), lecz znacznie gorzej zaprojektowano efekty eksplozji. Zawsze widać tylko duży krąg wybuchu. Nie rozumiem, dlaczego firma znana z pięknego programu demonstracyjnego „Fujiboink” potraktowała „Gato” po macoszemu. Z pewnością można było zrealizować to o wiele lepiej.

Pomimo to gra może się podobać, szczególnie na wyższych poziomach, gdzie trzeba się także zająć strategią. Po wykonaniu każdego zadania można natychmiast wywoływać nowe, aż do zatonięcia okrętu z powodu ciężkich uszkodzeń lub

przekroczenia dopuszczalnej głębokości zanurzenia. Prace remontowe są oczywiście możliwe, lecz tylko przy okręcie warsztatowym w bazie. Dodatkowym kłopotem są w takim wypadku trudne manewry cumowania do warsztatowca.

„Gato” nie dorównuje wprawdzie klasycznej pozycji tego gatunku — „Silent Service”, lecz również może dać graczowi wiele zadowolenia i przyjemności.

(ziew)

Producent: Atari Corporation

OCENA (w skali 1-10):

Grafika	7
Dźwięk	6
Realność	7
Sens gry	8



ARCHON

XL/XE/ST

Szachy są grą bardzo starą i z dużymi tradycjami. Autorem „Archona” nie przeszkodziło to jednak w zmodyfikowaniu tej gry i to tak bardzo, że powstała zupełnie nowa.

Przed wszystkim plansza została zwiększona o jeden rząd i jedną kolumnę pól i jej rozmiar wynosi 9x9. Ponadto plansza nie jest statyczna, lecz zmienia swoje właściwości w toku gry. Sygnalizowane jest zmianą jasności pól, które na przemian ciemnieją i jaśnieją (choć pola „czarne” zawsze pozostają ciemniejsze od „białych”). Plansza jest bowiem również ważnym elementem gry — od stopnia jasności pól, na którym walczą dwie figury, zależy siła figur (kolor jasny sprzyja figurom „białym”, zaś ciemny „czarnym”).

Figury w „Archonie” niczym nie przypominają zwykłych figur szachowych, a obaj gracze mają odmienne ich zestawy. Przedstawiają one różne baśniowe stwory, jak np. Trolle, Smoki oraz — oczywiście — czarodzieja i czarownicę, które pełnią w grze rolę przypominającą połączenie króla i hetmana w zwykłych szachach. Jeden z graczy kieruje siłami Światła, drugi zaś Ciemności (jest to doskonały pretekst do napisania legendy dla programu). Z tego powodu „białe” figury reprezentują stwory pozytywne, a „czarne” negatywne. Pomimo różnych figur obaj gracze dysponują taką samą siłą, gdyż zarówno globalna siła wszystkich figur, jak i moc oraz możliwości figur odpowiadających sobie są równoważne.

Dodatkową atrakcją jest sposób rozgrywania walki figur. W przeciwieństwie do szachów,

gdzie wynik starcia zależy od tego, kto pierwszy atakuje, może się tu zdarzyć wszystko. Gdy na jednym polu spotkają się figury dwóch graczy, następuje przeniesienie ich na drugą planszę i tam dopiero rozstrzygany jest wynik pojedynku. W ten sposób, oprócz siły samych figur i wpływu zaciemnienia lub rozjaśnienia pola, na rezultat pracuje jeszcze gracz za pomocą joysticka. Figura, która wygrała pojedynek, najczęściej wychodzi z niego osłabiona odniesionymi ranami i przeważnie staje się wtedy łatwym łupem nawet dla figury normalnie od niej słabszej.

„Archon” jest przeznaczony dla jednego gracza walczącego z komputerem lub dwóch graczy będących przeciwnikami. Ponadto można wybrać wariant

„Cyborg”, w którym walkę na głównej planszy toczy człowiek, a w pojedynkach figur zastępuje go komputer.

Niewątpliwie jest to gra godna polecenia wszystkim, dla których komputer jest nie tylko pomocą w łamaniu joysticka. Trochę gimnastyki przy pojedynkach z pewnością nie zaszkodzi nawet jajogłowemu. Dodatkowym argumentem przemawiającym za tym, że gra jest warta zainteresowania, jest fakt wyprodukowania drugiej wersji — „Archon II: Adept”.

(ziew)

Producent: Electronic Arts

OCENA (w skali 1-10):

Idea	10
Grafika	8
Dźwięk	6
Sens gry	9



W 1975 roku prototypowy Hughes YAH-64 wygrał konkurs na śmigłowiec szturmowy dla USA, pokonując konkurencyjny śmigłowiec Bell YAH-63. Od 1984 roku jako Hughes AH-64 Apache używany jest przez lotnictwo wojsk lądowych Stanów Zjednoczonych. Swojej użyteczności dowiódł już wielokrotnie na wielu poligonach i polach bitew. Ostatnio np. operacja Pustynna Burza rozpoczęła się od (skutecznego) ataku komandosów wspieranych przez AH-64A na irackie stanowiska wczesnego ostrzegania w Kuwejcie. Napędzany przez dwa turbowalowe silniki o mocy 1536 KM każdy, osiąga prędkość maksymalną 309 km/h, unosząc w powietrze około 1500 kg uzbrojenia.

„Gunship”, kolejne arcydzieło programistów z MicroProse Simulation Software, pozwala nam (przy odrobinie dobrej woli i wyobraźni) zasiąść za sterami jednego z najlepszych śmigłowców szturmowych XX wieku. Wyposażony w celownik lasero-

wy i telewizyjny na podczerwień, samonaprowadzające się pociski powietrze-ziemia i powietrze-powietrze, a także sześciolufowe działko kalibru 30 mm na wysięgniku pod kadłubem (kto oglądał „Błękitny Grom”, wie, co może zdziałać jego sekundowa salwa) zdecydowanie nie daje równych szans wrogowi.

Naszym śmigłowcem sterujemy przy pomocy joysticka i klawiatury (30 klawiszy do zapamiętania). Akrobacje nie są dostępne — „Apaczem” nie można wykonać pętli czy beczki jak w „Tomahawk”, ale to, co można, w zupełności wystarczy. Zaczynamy od kilku lotów szkolnych (Flight Training in the USA). „Wróg” strzela do nas ślepymi nabojami, odwzajemniamy mu się podobnie. Kiedy poczujemy się nieco pewniej, można iść na wojnę — autorzy programu polecają na początek Azję Południowo-Wschodnią, gdzie wróg jest słabo uzbrojony, ale trudny do zlokalizowania. Następny

stopień to Ameryka Środkowa — tu trzeba się już nieźle pilnować, jeśli chce się wrócić bezpiecznie do bazy. Dla znudzonych życiem przewidziano Bliski Wschód, gdzie Arabowie prują do nas z rakiet i dział przeciwlotniczych, a nad pustynią widać nas jak na dłoni. Na koniec — ostatni scenariusz, który oby nigdy się nie ziścił — desperackie akcje, aby powstrzymać Armię Czerwoną prącą przez Europę Zachodnią.

Do wyboru jest kilka stopni ryzyka. Dla początkujących najlepsze będą Regular Missions — regularne loty „według rozkazu”, gdzie ryzyko niewielkie, ale i o sławę trudno. Dla tych, którzy chcą się wykazać — Volunteer Missions, misje ochotnicze (na zasadzie „zostaliście wyznaczeni na ochotnika” — nie mamy wyboru) — ryzyko nieco większe, awanse nieco szybsze. I w końcu coś dla kandydatów na bohaterów — Volunteer Hazardous Duty, czyli „ryzykowne zadania ochotnicze”. Ze wszyst-

kiego, a także flary (FLARES) i pojemniki ze skrawkami folii aluminiowej (CHAFF) do ogłupiania wykrywaczy podczerwieni i radarów.

Najbardziej uniwersalny zestaw to dwa razy cztery rakiety „Hellfire”, dwa zasobniki po 19 pocisków FFAR i dwie rakiety „Sidewinder”. Ciekawa rzecz — z działka możemy zniszczyć każdy cel (nawet T-72 czy bunkier). Jest to tylko kwestia dystansu i amunicji (a ta szybko się kończy).

Po krótkiej praktyce wypracowałem sobie dość skuteczną taktykę: podejście z maksymalną prędkością tuż nad ziemią i błyskawiczny atak z odległości pół mili. Wystarczy krótka seria, a na małej wysokości wróg wykrywa nas, kiedy jest już za późno. Wymaga to skupionej uwagi, bo poniżej 500 stóp (a leci się na 150—300) śmigłowiec zaczyna „tańczyć” w górę i w dół.

Pilotaż „Apacza” jest dość łatwy — ale tylko sprawny „Apacz”. Za efekty specjalne towarzyszące rozmaitym uszkodzeniom autorom należy się duża premia. Awaria zawieszenia rotora lub napędu wirnika ogonowego powoduje, że śmigłowiec zatacza się jak pijany, a zachowanie kierunku — nie mówiąc już o lądowaniu — staje się dużą sztuką. W przypadku uszkodzenia jednego z silników mamy szansę utrzymać się w powietrzu po wystrzeleniu większej części amunicji. Jeśli nie zdążyliśmy tego zrobić, pozostaje tylko czekać na Mi-24 i wpakować w niego ile się da ołowiu.

Gdy dolatujemy do innej bazy niż ta, z której wystartowaliśmy, musimy podać odzew w odpowiedzi na hasło. Po trzeciej pomyłce przekonujemy się na własnej skórze o poziomie wyszkolenia amerykańskich artylerzystów.

Misję kończymy na cztery sposoby: roztrzaskując się o ziemię w podziurawionym kuli waku, dostając się do niewoli albo wracając przez linię frontu po awaryjnym lądowaniu na terenie wroga, ewentualnie ładując bezpiecznie w jednej z baz i decydując się na zakończenie misji. Jeśli spisaliśmy się dobrze, następuje najprzyjemniejszy moment gry: awanse i odznaczenia. Jeśli już nie żyjemy, musimy zaczynać karierę pilota od początku.

Programiści z MicroProse postarali się, aby nie było dwóch identycznych lotów. Nocne misje, ataki na składy amunicji, bazy, przeciwlotnicze, awarie podzespołów i inne atrakcje powodują, że odszedłem na emeryturę dopiero w stopniu pułkownika. Pisząc tę recenzję czuję jednak, że chyba wrócę do służby czynnej. I to jest istotna przewaga symulacji nad życiem.

Wojciech Setlak

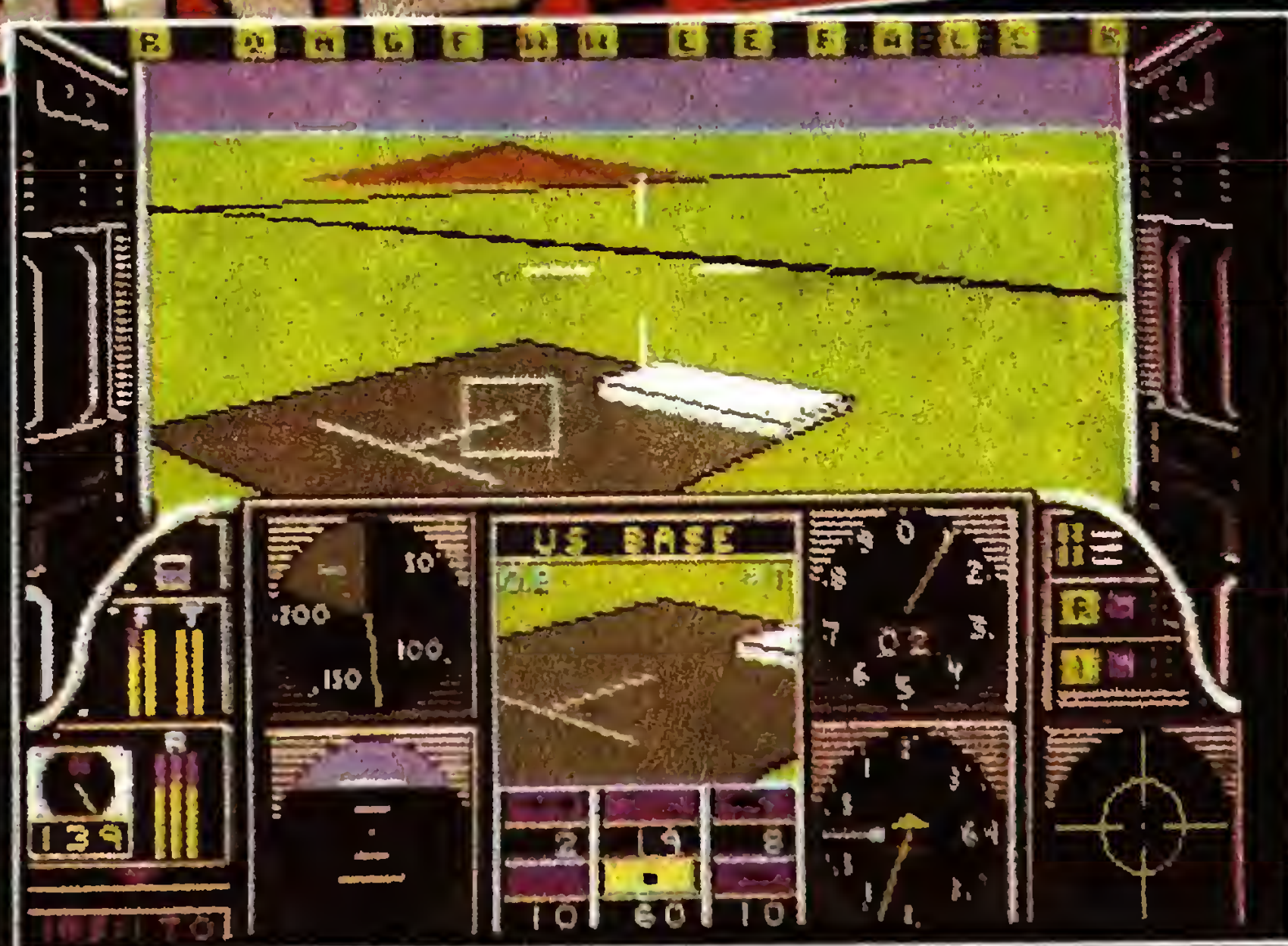
Producent: MicroProse Simulation Software

OCENA (w skali 1-10):

Grafika	8
Dźwięk	7
Realność	9
Sens gry	10

GUNSHIP

ST



kich stron sypie się deszcz medali, awansów i pocisków.

Poziom realizm symulacji jest wysoki. Jeśli chce się uniknąć mocnych wrażeń, można go obniżyć na kilka sposobów — ale chyba nie o to chodzi miłośnikom symulacji. Można także ustalać kwalifikacje przeciwnika — od elitarnych jednostek do przerażonych rezerwistów.

Przed startem dobrze jest zapisać sobie cele akcji (główny i dodatkowy), kierunek i siłę wiatru oraz temperaturę, która ma zasadniczy wpływ na udźwig, a zatem i na ilość uzbrojenia, jaką możemy zabrać. Poza tym możemy spotkać się z oficerem wywiadu (See Intel), który zapozna nas z poziomem wyszkolenia wroga, środkami obrony przeciwlotniczej, na jakie natkniemy się po drodze, a także uprzedzi nas o obecności śmigłowców Mi-24 w rejonie akcji. Możemy też spojrzeć na mapę okolicy, gdzie migają oznaczenia celów. Jeśli obleciał nas strach, idziemy do lekarza po L4 (Sick Call) i jesteśmy zwolnieni. Niestety, dowództwo bardzo tego nie lubi i każdy taki wypadek wpisuje nam skrupulatnie do akt, a to bardzo utrudnia awanse...

Tu parę uwag o doborze uzbrojenia. AH-64A wyposażony jest w działko 30 mm, cztery zaczepy podskrzydłowe i dwa na końcach skrzydeł (rzecz jasna „skrzydła” służą wyłącznie do podwieszania uzbrojenia). Do wyboru mamy: kierowane laserowo przeciwpancerne pociski AGM-114A „Hellfire” w zasobnikach po 4, samonaprowadzające się pociski klasy powietrze-powietrze AIM-9L „Sidewinder” oraz niekierowane, ale bardzo uniwersalne pociski raketowe FFAR kalibru 2.75 cala w zasobnikach po 7 i po 19 sztuk. Zabieramy 1200 sztuk amunicji do

STARGLIDER

ST

Z prawego ramienia spiralnego galaktyki nadlecieli zli Egronowie z zamiarem podbicia planety Novenia. Na szczęście Novenianie posiadają dobry system bezpieczeństwa złożony z Sentineli. Te olbrzymie statki kosmiczne chronią planetę i niszczą wszystko, co stanie na ich drodze.

Sentinele nie atakują tylko Stargliderów — ptaków z bardzo osobliwej rasy. Egronowie zbudowali więc flotę statków kosmicznych zbliżonych wyglądem do Stargliderów i wyprowadziwszy w pole Sentinele uderzyli na Novenię. Ostatnią szansą obrony Novenii jesteś Ty i Twój AGAV — uzbrojony po zęby ścigacz.

Idea „Starglidera” bardzo przypomina „Skyfox” firmy Electronic Arts. Gracz musi tak szybko, jak to możliwe zniszczyć statek macierzysty napastników, walcząc przy tym z licznymi, silnymi przeciwnikami. Ponadto trzeba wykonywać skomplikowane manewry, aby zacumować w hangarze remontowym celem uzupełnienia energii. Tak przedstawia się cała akcja gry — w zasadzie jest więc to typowa strzelanina.

Interesująca jest natomiast oprawa gry. Szczególne wrażenie sprawia bardzo szybka, trójwymiarowa grafika wektorowa w całości przeliczana przez procesor. Poszczególne obiekty nie tylko błyskawicznie przemykają po ekranie, lecz także są częściowo animowane. Na przykład, „Chodziarz” jest potężną maszyną bojową, która kroczy na dwóch nogach. Dźwięk również jest znakomity: digitalizowana muzyka wprowadzająca oraz efekty wybuchów i inne odgłosy nie wyczerpują wprowadzie wszystkich możliwości ST, lecz znacznie poprawiają atmosferę gry.



Szczególnie godne pochwały jest to, że „Starglider” działa zarówno na monitorze kolorowym, jak i czarno-białym, co jest bardzo rzadkim przypadkiem wśród gier przeznaczonych dla ST. Nie można go jedynie uruchomić na komputerach, które mają TOS odczytywany z dyskietki. Chociaż „Starglider” jest umieszczony na dyskietce, to jednak nie można zapisać na niej listy najlepszych wyników.

Dokumentacja „Starglidera” jest ważna i równocześnie dowcipna. Podręcznik i legenda zawierają nie tylko instrukcję obsługi, lecz także liczne sztuczki, bez których pozostaje ukryta finezja gry. Dodatkowy atut dokumentacji sta-

nowi lekki, humorystyczny język, jakim jest ona napisana.

„Starglider” jest znakomitą grą zręcznościową, technicznie prawie doskonałą i wykorzystującą możliwości komputera. Jeżeli więc lubisz trudne, lecz efektowne gry zręcznościowe, to „Starglider” z pewnością Ci się spodoba.

(ziew)

Producent: Rainbird

OCENA (w skali 1-10):

Grafika	8
Dźwięk	9
Sens gry	7

Wielki Konkurs!

Zapraszamy na trzecie wydanie nieustającego konkursu dla czytelników naszego pisma. Jak poprzednio do wygrania są następujące akcesoria ufundowane przez firmę „TAL”

- 1 joystick SV 127
- 10 joysticków SV 120
- 5 joysticków SV 119
- 3 wycinarki do dyskietek 5,25"
- 2 paczki po 10 dyskietek 5,25"
- 1 pudełko na dyskietki 5,25"

W konkursie może wziąć udział każdy, kto zechce odpowiedzieć na kilka pytań i wypełnić dokładnie zamieszczoną niżej ankietę. Nie zależy nam, aby pytania konkursowe były

trudne, lecz udzielenie poprawnych odpowiedzi będzie wymagało dokładnego przeczytania bieżącego numeru „Mojego Atari”.

Pod każdym pytaniem znajdują się trzy odpowiedzi, lecz tylko jedna z nich jest prawidłowa. Wystarczy ją podkreślić. Ci, którzy nie chcą niszczyć pisma wycinaniem, mogą odpowiedź zapisać na oddzielnej kartce w formie „1-a, 2-c itd”.

1 Jaka jest częstotliwość zegara taktującego procesor w konsoli Atari Lynx?

- 4 MHz;
- 8 MHz;
- 16 MHz.

2 Na jakiej dyskietce można zapisać ponad 1000 rozdań do gry „Blue Team Bridge”?

- na oryginalnej dyskietce BTB;
- na dowolnej dyskietce sformatowanej w rozszerzonej gęstości;
- na specjalnie przygotowanej dyskietce.

3 Plik określający parametry drukarki dla edytora 1st Word nosi nazwę:

- INSTALL.PRQ;
- PRINTER.HEX;
- PRINTER.CFG.

4 Jak nazywa się firma, która patronowała próbie bicia rekordu świata w długości grania na komputerze?

- Reline Software;
- First Star Software;
- Broderbund Software.

5 Jaką nazwę nosi komputer TT w wersji „wieżowej”?

- Tower TT;
- Telra TT;
- Manhattan TT.

ANKIETA

Imię i nazwisko:

dokładny adres:

posiadany komputer: Atari 800XL, 65XE,

130XE, 520ST 1040ST, inny:

rozmiar pamięci RAM komputera:

stosowana pamięć masowa: magnetofon,

turbo, stacja dysków

wiek: lat.

Uwaga!

Odpowiedzi przyjmujemy do dnia ukazania się kolejnego numeru „Mojego Atari”.

NOWY REKORD GUINNESSA



Jörg Kopmann, najwytrwalszy gracz świata.

Gra „Fate”, drugi bohater 120-godzinnej rekordu.

W normalnych warunkach trudno jest wysiedzieć przed komputerem dłużej niż kilka godzin. Zmęczenie powodowane ciągłym wpatrywaniem się w ekran i wysiłek umysłowy dają się we znaki nawet najwytrwalszym fanatom elektronicznej maszyny.

Ale są ludzie, którzy z łatwością pokonują barierę niedostępną dla przeciętnego śmiertelnika. Jednym z nich jest 20-letni Jörg Kopmann. Ten młody człowiek był w dniach od 4 do 9 marca głównym aktorem komputerowego maratonu. Jörg przez 120 godzin zmagał się z bohaterami gry przygodowej „Fate — Gates of Dawn”. Rezultat okazał się o 17 godzin lepszy od poprzedniego wyniku zapisanego w Księdze Guinnessa. Próba bicia rekordu odbyła się w pomieszczeniach biurowych firmy software'owej Reline Software, producenta wspomnianej gry „Fate”.

Z nowym rekordzistą przeprowadziła wywiad redakcja czasopisma ST Magazin. Jörg zapytany o samopoczucie od-

powiedział, że mógłby posiedzieć przy komputerze jeszcze parę godzin. Komputer jest dla niego wszystkim. Jeśli dano by mu do wyboru piękną dziewczynę, nawet miss Niemiec, i komputer, wołałby to drugie.

Do próby bicia rekordu Jörg wybrał grę przygodową, ponieważ obawiał się, że gra zręcznościowa będzie bardziej stresująca. Poza tym ciągłe naciśkanie przycisku fire joysticka przez tak długi czas grozi nabawieniem się odcisków i ran na kciuku. Problem wystąpił także w grze „Fate”. Mysz, która z konieczności aktywnie uczestniczyła w akcji, nie wytrzymała ciągłego przyciskania klawisza i organizatorzy musieli wymienić ją na nową. Po 120 godzinach zmagania grą „Fate” Jörg przyznał się, że znalazł zaledwie około 15% możliwych dróg rozwiązania.

Próba odbywała się pod kontrolą programistów z firmy Reline, którzy czuwaliby, żeby Jörg nie zasnął. Poza tym prawo do przeprowadzenia niespodziewanej inspekcji mieli kontrolerzy z redakcji Księgi Guinnessa.

W utrzymaniu dobrego samopoczucia i rzeźkości umysłu pomagały naszemu bohaterowi duże ilości soków owocowych i świeże bułeczki. Reguły bicia długotrwałych rekordów Guinnessa dopuszczają robienie co godzinę pięciminutowych przerw, które zawodnik może wykorzystać na krótką regenerację sił i zaspokojenie swoich potrzeb. Przerwy można oszczędzać i po pewnym czasie uciąć sobie jedną dłuższą, kilkunastominutową drzemkę.

Jörg Köpman wpisał się nie tylko na strony Księgi Guinnessa. Jego osobistym osiągnięciem było udowodnienie sobie, że potrafi grać dłużej, niż każdy inny fan komputerowych przygód. Obiecuje, że w przyszłości pokona każdego, kto spróbuje podwyższyć osiągnięty przez niego rekord. Następnym razem wytrwa tak długo, aż nikt już nie odważy się wyzwać do do komputerowego pojedynku.

Janusz JARMOCH

OSKAR

computer studio

04-118 Warszawa ul. Ostrobramska 128 tel. 100-061 w. 203 fax: 659-30-40

oferuje w stałej sprzedaży wraz z fachową obsługą

KOMPUTERY
MONITORY
JOYSTICKI
STACJE DYSKÓW
CARTRIDGE
DRUKARKI

SKANERY
MYSZKI
PROGRAMY
LITERATURĘ
SERWIS
MODERNIZACJE

W naszym salonie znajdziecie Państwo wszystko co jest niezbędne do pracy z Waszym komputerem. Mamy za sobą już 5 lat doświadczeń w pracy z domowymi komputerami i znamy ich możliwości i tajemnice.

Pomożemy Wam w ich rozwikłaniu i odkryjemy nieznane dotąd możliwości zastosowań.

Dla klientów z poza Warszawy prowadzimy działalność wysyłkową.

**Najtańsze
w Polsce!**



**SV 119
Junior**
2 Fire
6 Blaszanych styków
Prosty mechanizm



**SV 120
Junior-Stick**
2 Fire
6 Blaszanych styków
Uchwyt pistoletowy



**SV 122
Quickjoy II**
2 Fire
6 Blaszanych styków
AutoFire
Drażek lotniczy



**SV 124
Turbo**
6 Mikrostryków
AutoFire
Drażek lotniczy



**SV 123
Supercharger**
2 Fire
6 Mikrostryków
Ergonomiczna budowa
Precyzyjny mechanizm



**SV 126
Jet Fighter**
2 Fire
6 Mikrostryków
AutoFire
ACS-Regulator
szybkości AUTO
Obsługa pod kciuk"
Drażek lotniczy



**SV 125
Superboard**
6 Fire
10 Mikrostryków
AutoFire
Cyfrowy wyświetlacz
czasu
Sygnał dźwiękowy
Przełącznik dla
leworęcznych
Drażek lotniczy



**SV 130
IR Infrared**
1 Fire
5 Mikrostryków
Podczerwień
Daleki zasięg
Odbiornik



**SV 128
Megaboard**
4 Fire
10 Mikrostryków
AutoFire
6 cyfrowy stoper
ATM — Anti Tilt Mechanism
Fire Pad

**SV 140
Enterprise**

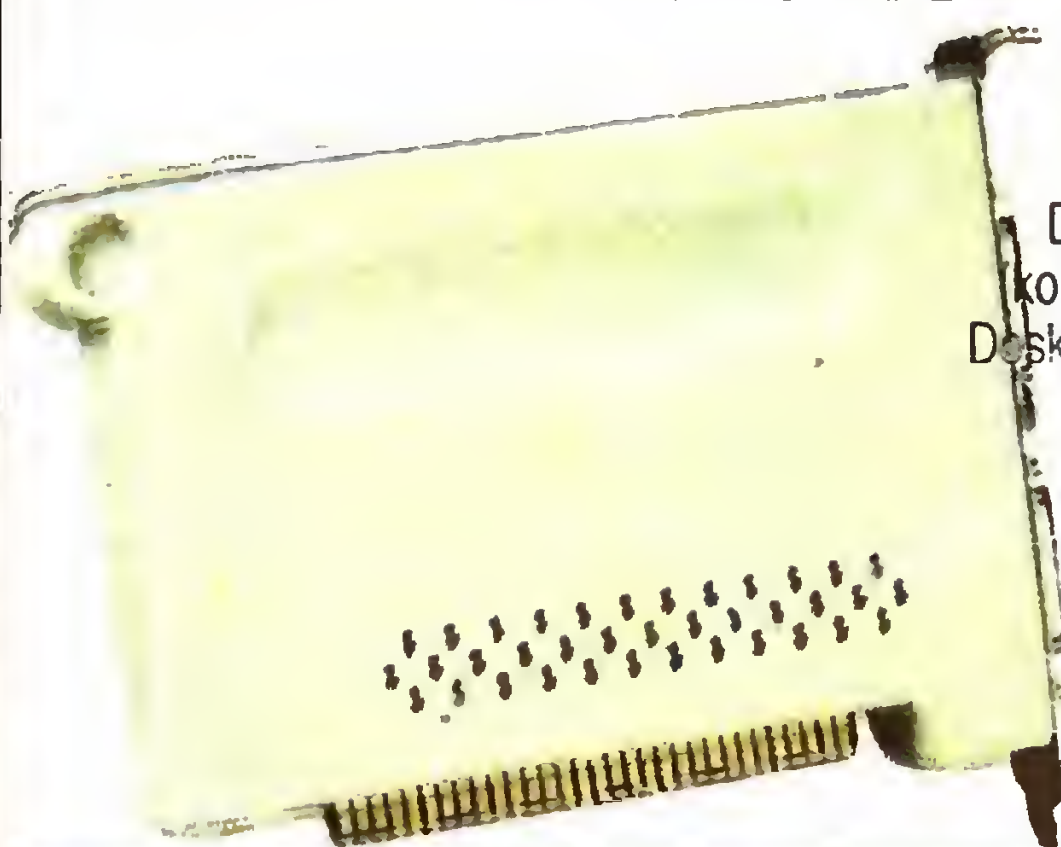
2 Fire
6 Mikrostryków
AutoFire
ACS — Regulator
szybkości AUTO
Drażek lotniczy
„kierownica”
Kabel 4 m



**SV 201
Quickjoy M 5**
Do IBM XT/AT
(kompatybilnych)
Współpracuje z Game-Card
lub I/O Card
2 Fire
2 AutoFire
6 Mikrostryków
Wybór AUTO
PSC — Regulator XY
Sygnalizacja Świetlna
Fire
ASC — Regulator szybkości
AUTO



**SV 202
M 6 analog**
Analogowy
DO IBM XT/AT
(kompatybilnych)
Współpracuje z Game-Card
lub I/O Card
2 Fire



**SV 210
Game Card**
Do IBM XT/AT
(kompatybilnych)
Doskonale pracuje
z M 5 i M 6



**SV 127
Top Star**
2 Fire
6 Mikrostryków
AutoFire
Przeźroczysta obudowa
SAS — Shock Absorbing
System
Platynowane części



**SV 500
Van 3**
Pudełko na dyskietki
80 sztuk 3 1/2"
Zamknięcie na klucz

**SV 510
Van 5**

Pudełko na dyskietki
80 sztuk 5 1/4"
Zamknięcie na klucz



SV 119 — 64 900
SV 120 — 74 900
SV 122 — 89 900
SV 124 — 109 900

SV 123 — 119 900
SV 125 — 239 900
SV 126 — 169 900
SV 127 — 249 900

SV 128 — 319 000
SV 130 — 339 000
SV 201 — 249 000
SV 202 — 229 000

SV 210 — 219 000
SV 500 — 119 000
SV 510 — 119 000

**interwencje
tel. 23-92-21**

SV 201 + SV 210 — 439 000
SV 202 + SV 210 — 399 000

Należy przesłać pieniądze przekazem zwykłym (czerwony blankiet) na adres:

TAL

Quickjoy

**Uwaga! W miejscu na
korespondencję prosimy podać
po raz drugi: adres domowy, kod
i symbol zamówionej pozycji.
W przypadku braku adresu
zwrotnego nie realizujemy
zamówienia.**

„TAL” — QWERTY
ul. Mikowa 45
02-411 Warszawa
fax: 659-12-35

Czas realizacji 14 dni

1st WORD I D R U K A R K I

Najpopularniejszym edytorem tekstu dla Atari ST jest **1st Word Plus**. W Polsce jest on użytkowany w wersji pozwalającej na uzyskanie na ekranie polskich liter. A co z drukarką? Bywa różnie.

1st Word Plus ma zwykle zainstalowane procedury drukujące po polsku na drukarkach Star. Wprawdzie nie zawsze chcą one poprawnie działać z innymi drukarkami, ale często są używane, gdyż użytkownicy nie potrafią przystosować programu do posiadanej drukarki. A przecież jest to bardzo proste.

1st Word Plus po uruchomieniu poszukuje na dyskiecie (w folderze, z którego został odczytany) pliku o nazwie **PRINTER.CFG**. Plik ten zawiera informacje o drukarce, które są następnie wykorzystywane przy wydruku. Każdy posiadacz drukarki może we własnym zakresie przygotować taki plik i to w ten sposób, aby jak najlepiej wykorzystać możliwości drukarki. Można nawet przygotować kilka takich plików przeznaczonych do drukowania różnego rodzaju dokumentów.

Pliki konfiguracyjne drukarek przygotowuje się przy użyciu samego **1st Word** z wyłączoną funkcją **EDYTOR TEKSTOWY**. Przygotowany plik należy następnie przetworzyć (skompilować) za pomocą pomocniczego programu **INSTALL.PR**. Posiadanie tego programu jest jedynym warunkiem, jaki musi być spełniony, aby samodzielnie wykonać plik konfiguracyjny.

Program **INSTALL.PR** znajduje się zwykle w folderze **INSTALL**. Ponadto folder ten zawiera przykładowe pliki konfiguracyjne, których nazwy mają rozszerzenie **.HEX**. Trzy najważniejsze z nich to **DAISY.HEX**, **MATRIX.HEX** i **STANDARD.HEX**. Zawierają one odpowiednio dane do skonfigurowania drukarek znakowych, mozaikowych oraz drukarek, które posiadają tylko możliwość drukowania standardowych znaków ASCII. W zależności od posiadanej przez nas drukarki jeden z tych plików będzie stanowił podstawę utworzenia własnego pliku konfiguracyjnego.

Budowa pliku HEX

Plik **.HEX** składa się z czterech części. W każdej z nich zapisuje się w określony sposób informacje dotyczące drukarki. Ewentualne komentarze można także zapisywać poprzedzając je gwiazdką. Każda informacja o drukarce musi być zapisana w oddzielnym wierszu, a dane liczbowe muszą być oddzielone przecinkami i podane w postaci szesnastkowej (hexadecymalnej).

Pierwszą informacją jest nazwa drukarki wzbogacona ewentualnie o charakterystyczny parametr pliku konfiguracyjnego. Cały napis nie może mieć długości większej niż 32 znaki. Na przykład, plik konfiguracyjny pokazany na listingu 1 zawiera nazwę „Citizen 120D standard”, a jego wersja poprawiona (będzie to opisane dalej) „Citizen 120D download”. Można w ten sposób uwidocznic różne parametry, jak choćby rodzaj papieru: „kartki” lub „rolka”.

Kolejna grupa danych zawiera sześć wartości, które określają szczegóły konfiguracji wymagane przez **1st Word**. Wartości te muszą być zapisane w jednym wierszu pliku, a są one następujące:

- 1 — typ drukarki
- 2 — szerokość znaku
- 3 — wysokość wiersza
- 4 — gęstość punktów
- 5 — odstęp pogrubienia
- 6 — przerwa między stronami

Posiadacze drukarek matrycowych (igłowych), które stanowią przeważającą większość, w punktach 1, 2 i 5 wpisują zero. Dla drukarek znakowych pierwsza dana musi mieć wartość 1, druga określa szerokość znaku w jednostkach drukarki, a piąta wielkość przesunięcia głowicy dla uzyskania druku pogrubionego (odpowiednie wartości można znaleźć w instrukcji drukarki). Trzecia dana zależy od pionowej rozdzielczości drukarki i wynosi (przeważnie) — sprawdzić w instrukcji) \$0C dla drukarek 9-igłowych i \$0A dla 24-igłowych.

Czwarta wartość określa gęstość matrycy znaku w punktach na cal (pixel per inch) i jest równa dla większości drukarek 9-igłowych \$48, a dla 24-igłowych \$3C. Ostatnia (szósta) dana zależy wyłącznie od rodzaju stosowanego papieru. Jeśli jest ona równa 1, to program zatrzymuje drukowanie po zakończeniu każdej strony. Gdy wartość ta jest równa 0, cały dokument jest drukowany bez przerw.

Trzeci blok danych określa charakterystykę drukarki. Jest on zbudowany w formie tabeli. Pierwszą liczbą w każdym wierszu jest numer pozycji tabeli, a po nim następują kody, które trzeba przekazać drukarce w celu zrealizowania funkcji odpowiadającej tej pozycji. Jeżeli funkcja wymaga podania liczby będącej parametrem, a nieznanej w chwili tworzenia pliku (na przykład, długość strony w wierszach, która jest ustalana podczas pisania dokumentu), to zamiast tej liczby wpisuje się wartość \$80. Wartość ta będzie podczas druku odpowiednio zmieniona przez **1st Word**. Funkcje niedostępne na posiadanej drukarce należy po prostu pominąć, lecz kolejność funkcji w pliku nie może być zmieniana. Koniec bloku opisującego drukarkę musi być oznaczony wierszem zawierającym pojedynczą wartość 0.

Nie będę tu podawał kolejnych funkcji, gdyż są one opisane w instrukcji oraz przykładowych plikach **.HEX**. Funkcje te można ponadto odczytać z listingu 1 (są one podane w języku angielskim, aby ułatwić wyszukiwanie w instrukcji drukarki).

Ostatni blok pliku konfiguracyjnego jest tabelą zamiany znaków zestawu stosowanego w Atari ST na znaki dostępne w drukarce. Jego konstrukcja jest zbliżona do poprzedniego bloku i zawiera w każdym wierszu numer znaku z zestawu ST, a po nim kody przesyłane do drukarki w celu uzyskania tego znaku.

LISTING 1

```

.....
.....
MATRIX PRINTER CONFIGURATION
.....
.....

* Printer name
Citizen 120D standard

* Configuration variables
0,0,C,48,0,1

* Printer characteristics
1. A,D
4. 18,42,80,0,8
5. 18,45
7. 18,46
8. 18,46
9. 18,46
A. 18,34
B. 18,35
C. 18,34
D. 18,35
E. 18,78,1
F. 18,78,0
10. 18,78,1
11. 18,78,0
12. 18,53,0
13. 18,54
14. 18,53,0
15. 18,54
16. 18,53,1
17. 18,54
18. 18,53,1
19. 18,54
1A. 18,2D,1
1B. 18,2D,0
1C. 18,2D,1
1D. 18,2D,0
1E. C
1F. 12
20. 18,52,0
21. 18,40,1B,38
22. 0
23. 0
24. 18,43,80
25. 18,41,80
26. 18,2A,5
27. 18,78,0,1B,21,0
28. 18,78,0,1B,21,10
2A. 18,78,0,1B,21,1
2B. 18,78,0,1B,21,11
2C. 18,78,0,1B,21,4
2D. 18,78,0,1B,21,14
2E. 18,78,0,1B,21,20
2F. 18,78,0,1B,21,30
0

* Carriage return line feed
* Vertical tab to line
* Draft bold on
* Draft bold off
* NLQ bold on
* NLQ bold off
* Draft italic on
* Draft italic off
* NLQ italic on
* NLQ italic off
* Draft light on
* Draft light off
* NLQ light on
* NLQ light off
* Draft superscript on
* Draft superscript off
* NLQ superscript on
* NLQ superscript off
* Draft subscript on
* Draft subscript off
* NLQ subscript on
* NLQ subscript off
* Draft underline on
* Draft underline off
* NLQ underline on
* NLQ underline off
* Formfeed
* Horizontal initialization
* Vertical initialization
* Termination printer report
* Backspace
* Carriage return
* Form length in lines
* Set line feed to N units
* Start graphics (mode 5)
* Draft Pica (10 cpi)
* NLQ Pica (10 cpi)
* Draft Elite (12 cpi)
* NLQ Elite (12 cpi)
* Draft Condensed (17.2 cpi)
* NLQ Condensed (17.2 cpi)
* Draft Expanded (6 cpi)
* NLQ Expanded (6 cpi)
* NULL termination byte

Translation Table
7F  - nothing
80  - nothing
81. 18, 52, 2, 7D, 1B, 52, 0
82. 18, 52, 1, 7B, 1B, 52, 0
83. 01, 0, 5E
84. 18, 52, 2, 7B, 1B, 52, 0
85. 18, 52, 1, 4D, 1B, 52, 0
86. 18, 52, 4, 7D, 1B, 52, 0
87. 18, 52, 1, 5C, 1B, 52, 0
88. 05, 0, 5E
89. 05, 0, 1B, 52, 1, 7E, 1B, 52, 0
8A. 18, 52, 1, 7D, 1B, 52, 0
8B  - nothing
8C  - nothing
8D. 18, 52, 6, 7E, 1B, 52, 0
8E. 18, 52, 2, 5B, 1B, 52, 0
8F. 18, 52, 6, 5D, 1B, 52, 0
90. 18, 52, 9, 4D, 1B, 52, 0
91. 18, 52, 4, 7B, 1B, 52, 0
92. 18, 52, 4, 5B, 1B, 52, 0
93. 0F, 0, 5E
94. 18, 52, 2, 7C, 1B, 52, 0
95. 18, 52, 6, 7C, 1B, 52, 0
96. 75, 0, 5E
97. 18, 52, 1, 7C, 1B, 52, 0
98. 79, 0, 1B, 52, 1, 7E, 1B, 52, 0
99. 18, 52, 2, 5C, 1B, 52, 0
9A. 18, 52, 2, 5D, 1B, 52, 0
9B. 03, 0, 7C
9C. 18, 52, 3, 23, 1B, 52, 0
9D. 18, 52, 6, 5C, 1B, 52, 0
9E. 18, 52, 2, 7E, 1B, 52, 0
9F. 66
A0. 18, 7E, 35, 1, A0, 1B, 7E, 35, 0
A1. 18, 7E, 35, 1, A1, 1B, 7E, 35, 0
A2. 18, 7E, 35, 1, A2, 1B, 7E, 35, 0
A3. 18, 7E, 35, 1, A3, 1B, 7E, 35, 0
A4. 18, 52, 7, 7C, 1B, 52, 0
A5. 18, 52, 7, 5C, 1B, 52, 0
A6. 18, 7E, 35, 1, A6, 1B, 7E, 35, 0
A7. 18, 7E, 35, 1, A7, 1B, 7E, 35, 0
A8. 18, 52, 7, 5D, 1B, 52, 0
A9. 18, 7E, 35, 1, A9, 1B, 7E, 35, 0
AA. 18, 7E, 35, 1, AA, 1B, 7E, 35, 0
AB. 18, 7E, 35, 1, AB, 1B, 7E, 35, 0
AC. 18, 7E, 35, 1, AC, 1B, 7E, 35, 0
AD. 18, 52, 7, 5B, 1B, 52, 0
AE. 18, 7E, 35, 1, AE, 1B, 7E, 35, 0
AF. 18, 7E, 35, 1, AF, 1B, 7E, 35, 0
B0. 01, 0, 7E
B1. 6F, 0, 7E
B2. 18, 52, 4, 5C, 1B, 52, 0
B3. 18, 52, 4, 7C, 1B, 52, 0
B4  - nothing
B5  - nothing
B6  - nothing
B7  - nothing
B8  - nothing
B9. 18, 52, 1, 7E, 1B, 52, 0
BA  - nothing
BB  - nothing
BC  - nothing
BD  - nothing
BE  - nothing
BF  - nothing
C0. 8D
C1. A1, 0, 2C
C2. A3, 0, 27
C3. A5, 0, 2C
C4. AC, 0, 2F
C5. AE, 0, 27
C6. AF, 0, 27
C7. 53, 0, 27
C8. 5A, 0, 27
C9. 5A, 0, 2D
CA  - nothing
CB  - nothing
CC  - nothing
CD  - nothing
CE  - nothing
CF  - nothing
D0  - nothing
D1. 01, 0, 2C
D2. 03, 0, 27
D3. 05, 0, 2C
D4. 0C, 0, 2F
D5. 0E, 0, 27
D6. 0F, 0, 27
D7. 73, 0, 27
D8. 7A, 0, 27
D9. 7A, 0, 60
DA  - nothing
DB  - nothing
DC  - nothing
DD. 18, 52, 2, 4D, 1B, 52, 0
DE  - nothing
DF. 18, 7E, 35, 1, ED, 1B, 7E, 35, 0
E0. 18, 7E, 35, 1, ED, 1B, 7E, 35, 0
E1. 18, 52, 2, 7E, 1B, 52, 0
E2. 18, 7E, 35, 1, E2, 1B, 7E, 35, 0
E3. 18, 7E, 35, 1, E3, 1B, 7E, 35, 0
E4. 18, 7E, 35, 1, E4, 1B, 7E, 35, 0
E5. 18, 7E, 35, 1, E5, 1B, 7E, 35, 0
E6. 18, 7E, 35, 1, E6, 1B, 7E, 35, 0
E7. 18, 7E, 35, 1, E7, 1B, 7E, 35, 0
E8. 18, 7E, 35, 1, E8, 1B, 7E, 35, 0
E9. 18, 7E, 35, 1, E9, 1B, 7E, 35, 0
EA. 18, 7E, 35, 1, EA, 1B, 7E, 35, 0
EB. 18, 7E, 35, 1, EB, 1B, 7E, 35, 0
EC  - nothing
ED  - nothing
EE. 18, 7E, 35, 1, EE, 1B, 7E, 35, 0
EF. 18, 7E, 35, 1, EF, 1B, 7E, 35, 0
F0. 18, 7E, 35, 1, F0, 1B, 7E, 35, 0
F1. 18, 7E, 35, 1, F1, 1B, 7E, 35, 0
F2. 18, 7E, 35, 1, F2, 1B, 7E, 35, 0
F3. 18, 7E, 35, 1, F3, 1B, 7E, 35, 0
F4  - nothing
F5  - nothing
F6. 18, 7E, 35, 1, F6, 1B, 7E, 35, 0
F7. 18, 7E, 35, 1, F7, 1B, 7E, 35, 0

```



Występuje tu jednak jedna istotna różnica. Po numerze znaku ST nie trzeba wpisywać żadnych kodów, lecz w takim przypadku zamiast tego znaku wydrukowany zostanie odstęp. Natomiast pominięcie znaku w tabeli zamiany spowoduje bezpośrednie przesłanie tego znaku do drukarki. To ostatnie rozwiązanie stosuje się w przypadkach, gdy znak w zestawie drukarki wiernie odpowiada znakowi w zestawie ST.

Przygotowanie Pliku HEX

Przystępując do pracy uruchamiamy najpierw **1st Word** i wczytujemy wybrany plik .HEX. Jeśli takiego pliku nie ma na dyskiecie, to będziemy musieli go utworzyć samodzielnie korzystając z podanego tu wzoru oraz opisu zawartego w instrukcji programu. Następnie korzystając z instrukcji drukarki w odpowiednich miejscach wpisujemy szesnastkowo właściwe wartości kodów sterujących.

Przedtem trzeba jednak zastanowić się nad sposobem wykorzystania wszystkich możliwości drukarki i ominięcia jej ograniczeń. **1st Word** może drukować tekst czcionką zwykłą, grubą, jasną i pochyłą. Każdy dokument może być ponadto drukowany w trybie *draft* lub *NLQ*. Pomyślmy zatem, jak te możliwości dostosować do konkretnej drukarki. Przykładem niech będzie prosta drukarka Citizen 120D.

Drukarka ta umożliwia druk prosty lub pochyły (*kursywa*) czcionką zwykłą, wzmocnioną (*doublestrike*), pogrubioną (*emphasized* lub *bold*) i w jakości listowej (*NLQ*). Wszystkie te tryby druku można dowolnie mieszać. Niestety, w trybie *NLQ* nie można drukować znaków, których definicje znajdują się w pamięci RAM drukarki (zdefiniowane przez użytkownika). Z drugiej strony wszystkim wiadomo, że jakość druku mocno zależy od stopnia zużycia taśmy. Z tego względu druk wzmocniony (*doublestrike*) należy stosować, gdy taśma jest nieco zużyta.

Aby wykorzystać możliwości drukarki z uwzględnieniem podanych wyżej uwag przyjąłem następujące przypisanie wariantów druku programu wariantom druku drukarki:

zwykły	— zwykły (<i>draft</i>)
gruby	— pogrubiony (<i>emphasized</i>)
jasny	— <i>NLQ</i>
NLQ	— wzmocniony (<i>doublestrike</i>)

Takie przypisanie wymaga zamiany w pliku konfiguracyjnym kodów odpowiadających trybowi *NLQ* na kody trybu *doublestrike*, a kodów dla liter jasnych na kody dla trybu *NLQ* (patrz listing 1). W ten sposób można wykorzystać jasne litery, które w zamysle twórców **1st Word**a były przeznaczone do druku czerwonego na drukarkach kolorowych.

Nietypowe litery

W stanowiącej czwartą część pliku konfiguracyjnego tabeli zamiany kodów wpisujemy instrukcje nakazujące drukarce utworzenie odpowiednich liter. Niekiedy możliwe jest proste zastąpienie brakującego znaku podobną literą. Na przykład „f” (kod \$9F) drukowane jest „f”. W pozostałych przypadkach stosuje się trzy podstawowe rozwiązania:

LISTING 2

```

FB, 1B, 52, 1, 5B, 1B, 52, 0 * *
FD, 1B, 7E, 35, 1, FB, 1B, 7E, 35, 0 * *
FA * * nothing
FB, 1B, 7E, 35, 1, FB, 1B, 7E, 35, 0 * J
FC, 1B, 7E, 35, 1, FC, 1B, 7E, 35, 0 * A
FD, 1B, 7E, 35, 1, FD, 1B, 7E, 35, 0 * J
FE * * nothing
0 * NULL terminator

10 * Font Generator
20 * Wojciech Ziemiański
30 * (c) 1991, Sp. Bajtek
40 *
50 ? "Reading data" if $=""
60 read d
70 if d<>1 then f=f+chr(d):goto 60
80 ? "Creating file"
90 open "0", "a", "CIT120D.FNT"
100 ? f, f
110 close #1
120 end
997 *
998 * font data
999 *
1000 * reset
1010 data 27, 64
1020 * define 49-57
1030 data 27, 3B, 0, 49, 57
1040 * A
1050 data 0, 8, 20, 64, 20, 64, 20, 64, 5B, 5, 0, 0
1060 * C
1070 data 12B, 2B, 34, 0, 34, 64, 162, 0, 34, 0, 0, 0
1080 * E
1090 data 0, 5B, 6B, 15, 6B, 15, 6B, 15, 6B, 4B, 0, 0
1100 * I
1110 data 12B, 0, 0, 13B, 0, 254, 0, 34, 0, 0, 0, 0
1120 * N
1130 data 12B, 62, 0, 32, 0, 9B, 12B, 32, 30, 0, 0, 0
1140 * O
1150 data 12B, 2B, 34, 0, 34, 64, 162, 0, 34, 2B, 0, 0
1160 * S
1170 data 12B, 1B, 42, 0, 42, 64, 173, 0, 42, 4, 0, 0
1180 * Z
1190 data 12B, 0, 34, 4, 34, 13B, 34, 16, 34, 0, 0, 0
1200 *
1210 data 12B, 0, 34, 4, 34, 72, 1B2, 1B, 34, 0, 0, 0
1220 * define 35-39
1230 data 27, 3B, 0, 35, 39
1240 * A
1250 data 12B, 30, 32, 72, 12B, 8, 12B, 75, 32, 30, 0, 0
1260 * C
1270 data 12B, 60, 6B, 0, 9B, 0, 194, 0, 6B, 36, 0, 0
1280 * E
1290 data 12B, 254, 0, 14B, 0, 14B, 0, 14B, 1, 130, 0, 0
1300 * I
1310 data 12B, 4, 0, 254, 0, 1B, 0, 34, 0, 2, 0, 0
1320 * N
1330 data 12B, 12B, 0, 32, 1B, 8, 6B, 130, 0, 12B, 0, 0
1340 * define 60-62
1350 data 27, 3B, 0, 60, 62
1360 * O
1370 data 12B, 60, 6B, 0, 9B, 0, 194, 0, 6B, 6B, 0, 0
1380 * S
1390 data 12B, 0, 130, 20, 13B, 1B, 162, 80, 130, 0, 0, 0
1400 * Z
1410 data 12B, 36, 82, 0, 82, 12B, 82, 9, 82, 12, 0, 0
1420 * define 42
1430 data 27, 3B, 0, 42, 42
1440 * Z
1450 data 12B, 0, 6B, 36, 74, 12B, 82, 32, 6B, 0, 0, 0
1460 * end of data
1470 data -1

```

1. drukowanie znaku, a następnie cofnięcie głowicy i dodrukowanie dodatkowych elementów;
2. przełączenie drukarki na jeden z zestawów znaków narodowych, wydrukowanie znaku i ponowne przełączenie na zestaw standardowy;
3. przełączenie drukarki na zestaw znaków w standardzie IBM, wydrukowanie znaku i ponowne przełączenie na zestaw Epson.

Pierwszy sposób jest najprostszy do uzyskania, lecz zmniejsza prędkość druku. Stosuje się go zwykle tylko do znaków, których nie można uzyskać w inny sposób. W przykładowym pliku w ten sposób tworzona jest m.in. litera „ä”: najpierw drukuje się „a”, następnie cofa głowicę i dodrukowuje „~”. Na szczęście wiele liter znajduje się w narodowych zestawach znaków drukarki. Na przykład, Citizen posiada siedem takich zestawów. Tak powstaje m.in. li-

dokończenie na str. 31

LISTING 3

```

21, 1B, D, 1B, 32, 1B, 21, 0, 1B, 43, B, 1B, 4F, 1B, 65, 1, 1, 1B, 65, 0, 8,
1B, 6C, 0, 1B, 3B * Termination: printer reset

C1, 1B, 25, 1, 0, 23, 1B, 25, 0, 0 * A
C2, 1B, 25, 1, 0, 24, 1B, 25, 0, 0 * C
C3, 1B, 25, 1, 0, 25, 1B, 25, 0, 0 * E
C4, 1B, 25, 1, 0, 26, 1B, 25, 0, 0 * I
C5, 1B, 25, 1, 0, 27, 1B, 25, 0, 0 * N
C6, 1B, 25, 1, 0, 3C, 1B, 25, 0, 0 * O
C7, 1B, 25, 1, 0, 3E, 1B, 25, 0, 0 * S
C8, 1B, 25, 1, 0, 2A, 1B, 25, 0, 0 * Z
C9, 1B, 25, 1, 0, 3D, 1B, 25, 0, 0 * Z

D1, 1B, 25, 1, 0, 31, 1B, 25, 0, 0 * a
D2, 1B, 25, 1, 0, 32, 1B, 25, 0, 0 * c
D3, 1B, 25, 1, 0, 33, 1B, 25, 0, 0 * e
D4, 1B, 25, 1, 0, 34, 1B, 25, 0, 0 * i
D5, 1B, 25, 1, 0, 35, 1B, 25, 0, 0 * n
D6, 1B, 25, 1, 0, 36, 1B, 25, 0, 0 * o
D7, 1B, 25, 1, 0, 37, 1B, 25, 0, 0 * s
D8, 1B, 25, 1, 0, 39, 1B, 25, 0, 0 * z
D9, 1B, 25, 1, 0, 3B, 1B, 25, 0, 0 * z

```



Outline-Art jest programem służącym do tworzenia grafiki wektorowej. Podstawowym jego zastosowaniem jest przygotowywanie rysunków wektorowych dla Calamusa. Outline-Art wymaga monitora monochromatycznego oraz komputera ST o dowolnej pojemności pamięci.

MENU EKRANU

- zmniejszenie
- cała strona
- obszar do powiększenia
- powiększenie

MENU OBIEKTÓW

- tekst obrócony
- tekst na łuku
- tekst na wzorze
- wzór
- dynamiczna skala szarości
- wzór sterujący

MENU REDAGOWANIA OBIEKTÓW

- zamiana tekstu na wzór
- projekcja obiektu wg wzoru sterującego
- formularz parametrów obiektu
- dodawanie obiektu
- wybranie obiektu
- skopiowanie obiektu
- umieszczenie obiektu pod spodem
- umieszczenie obiektu na wierzchu
- skasowanie obiektu

MENU ELEMENTÓW

- linia prosta
- krzywa Bezierra
- wybór elementu

dokończenie ze str. 29

tera „ä”, która jest umieszczona w zestawie niemieckim (nr 2 = Germany) z kodem \$7B (normalnie jest to prawy nawias klamrowy — „}”).

Niemal wszystkie nowsze drukarki posiadają możliwość programowego przełączenia się w tryb emulacji drukarki IBM (np. Citizen) lub na korzystanie z zestawu znaków tej drukarki (np. Star). Można to wykorzystać do uzyskania dalszych znaków, które są inaczej nieosiągalne. W naszym przykładzie używane są tak wszystkie litery alfabetu greckiego.

W szczególnych przypadkach możliwe jest jeszcze łączenie opisanych sposobów. I tak otrzymanie litery „ÿ” wymaga wydrukowania „y”, cofnięcia głowicy, przełączenia na zestaw francuski (nr 1 = French) i dodrukowania „ ”.

Polskie litery w **1st Word Plus** są rozmieszczone według standardu Klubu Atari. Zajmują one miejsce znaków hebrajskich o kodach od \$C1 do \$C9 i od \$D1 do \$D9. Niestety, niewiele jest jeszcze drukarek, które mają zapisane wzory polskich liter. Z tego powodu jedynym sposobem ich uzyskania jest nadrukowywanie dodatkowych elementów. Tak otrzymane litery nie wyglądają jednak estetycznie i rozwiązanie to może być stosowane tylko w ostateczności.

Uczenie drukarki

Drukarki są obecnie wyposażone w pamięć RAM o pojemności co najmniej 4 KB. Jest to wystarczająco duża przestrzeń, aby zmieścić w niej dane osiemnastu polskich liter. Jeśli więc chcemy uzyskać ładne litery, to musimy nauczyć drukarkę ich drukowania. Oczywiście wygląd liter musimy opracować sami, chyba że dysponujemy gotowymi wzorami. Projektowanie liter jest opisane w instrukcji każdej drukarki i nie sprawia kłopotu, choć jest to zajęcie dość żmudne, szczególnie w przypadku drukarek 24-igłowych oraz liter w trybie NLQ.

Zaprojektowane litery trzeba jeszcze jakoś przekazać drukarce. Najprostsze — choć mało eleganckie — rozwiązanie polega na zapisaniu danych liter w oddzielnym pliku i wywołaniu tego pliku przed uruchomieniem **1st Word**a. Po wyświetleniu okna dialogowego GEM wybieramy odpowiedź „PRINT” i nasz plik jest przekazywany drukarce. Przykład programu tworzącego taki plik jest pokazany na listingu 2. Program ten jest napisany w języku ST Basic sprzedawanym razem z komputerem.

Użytkownicy, którzy potrafią trochę programować, mogą wykonać to bardziej elegancko (i wygodniej). Należy napisać program zapisujący dane polskich liter bezpośrednio na drukarce. Program ten następnie kompilujemy, aby uzyskać działającą wersję w języku maszynowym. Na koniec specjalnym programem przekształcamy program typu .PRG na program .ACC (accessory). W rezultacie otrzymujemy program, który instaluje się w pamięci podczas uruchamiania komputera i umożliwia przekazanie drukarce polskich liter w dowolnym momencie pracy.

Czy taki sposób uzyskiwania polskich liter ma wpływ na zawartość pliku konfiguracyjnego drukarkę? Oczywiście, przecież teraz będą one uzyskiwane przez przełączenie drukarki na druk według wzorów z pamięci RAM zamiast nadrukowywania. Odpowiednie zmiany w tabeli translacji znaków (dla zestawu tworzonego przez program z listingu 2) są pokazane na listingu 3.

Dodatkową zmianą jest inny sposób realizacji funkcji RESET drukarki (numer funkcji \$21). Funkcja ta jest wywoływana przez **1st Word** po zakończeniu druku każdego dokumentu. Normalnie zapisuje się tam kod RESET (jednakowy dla prawie wszystkich drukarek — \$1B,\$40). Przy korzystaniu ze znaków zdefiniowanych w RAM drukarki powoduje to jednak ich usunięcie z pamięci. Zamiast kodów RESET trzeba więc w tym miejscu wpisać kody wyłączania wszystkich wariantów pracy drukarki tak, aby ustawić ją w takim stanie, w jakim jest bezpośrednio po włączeniu zasilania (z wyjątkiem kasowania zawartości pamięci). Należy przy tym pamiętać, że wszystkie niezbędne kody muszą się znaleźć w jednym wierszu (ze względów technicznych na wydruku są w dwóch wierszach).

Opisany powyżej sposób stosuję z powodzeniem od kilku lat i nigdy nie miałem kłopotów z drukowaniem dokumentów przy pomocy programu **1st Word Plus**.

Wojciech ZIENTARA

1st Word i drukarki



Wieża do samodzielnego montażu

TETRA 030

Komputer odpowiadający ATARI TT

- * Procesor Motorola 68030
- * 8 MB RAM
- * Stacja dysków elastycznych 1,44 MB / 3,5"
- * Stacja dysków elastycznych 1,2 MB / 5,25"
- * Twardy dysk 213 MB
- * Komplet standardowych złącz

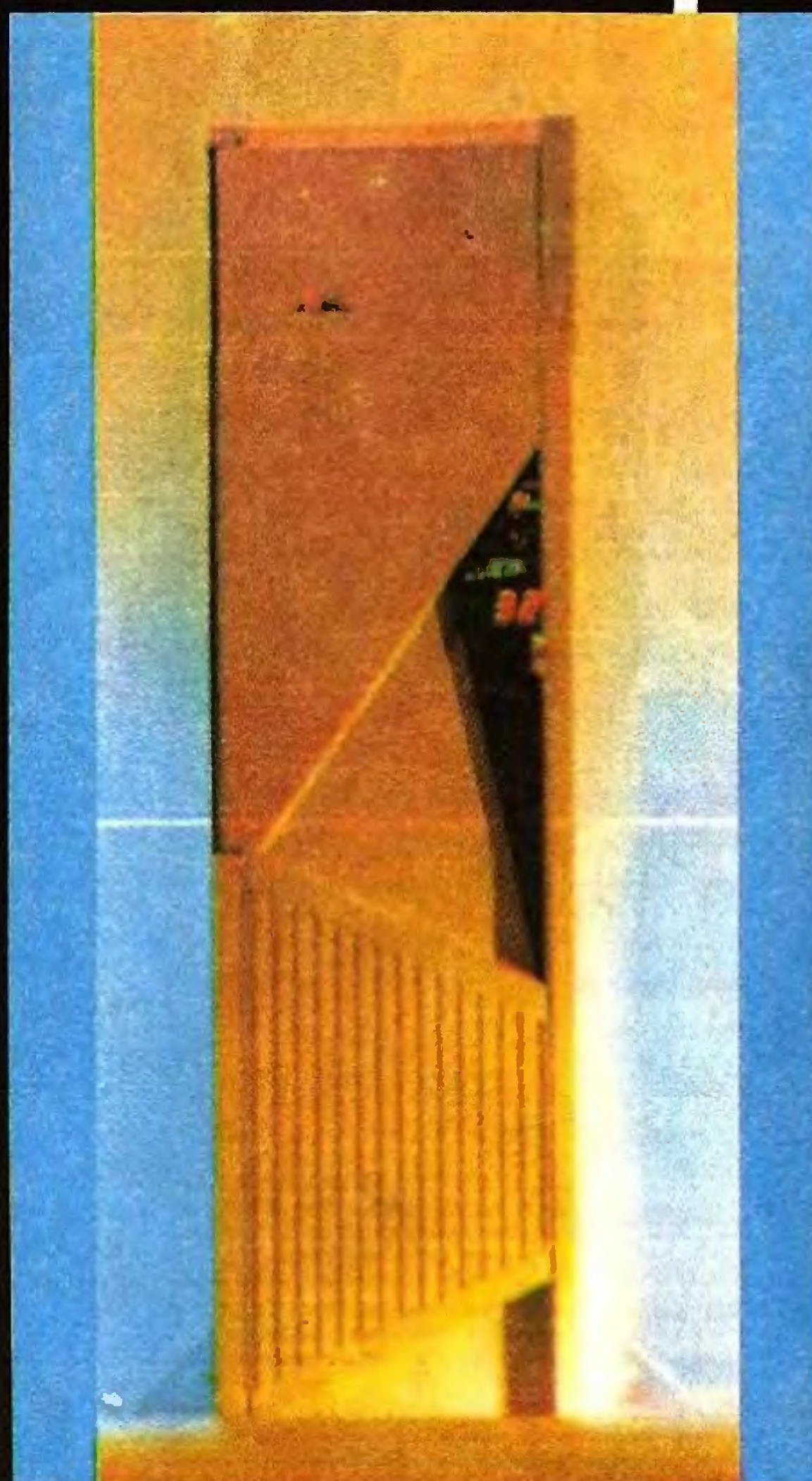
Wyposażenia dodatkowe:

- * Dysk wymienny 44 MB / 5,25"
- * Streamer 155 MB / 3,5"
- * Różne typy monitorów
- * Klawiatura zgodna ze standardami PC

Dokończenie ze str. 32

68030, taktowanego z częstotliwością 32 MHz. Jest on wyposażony w dwa napędy dysków elastycznych: 3,5 calowy o pojemności 1,44 MB i 5,25 cala o pojemności 1,2 MB. Oprócz tego użytkownik ma do dyspozycji szybki twardy dysk 213 MB o czasie dostępu 15 milisekund. Zasilacz Tetra 030 ma moc 220 W i jest chłodzony wentylatorem. Do wyposażenia komputera należy także mysz i wysokiej jakości klawiatura stosowana w komputerach typu PC, jednakże zgodna ze standardem oryginalnego TT.

Wieża TT nadaje się do zastosowań przemysłowych. Układy wyjścia zrealizowane w technice TTL oraz 8-bitowy analogowo-cyfrowy układ wejścia sprawiają, że można ją stosować w procesach sterowania.



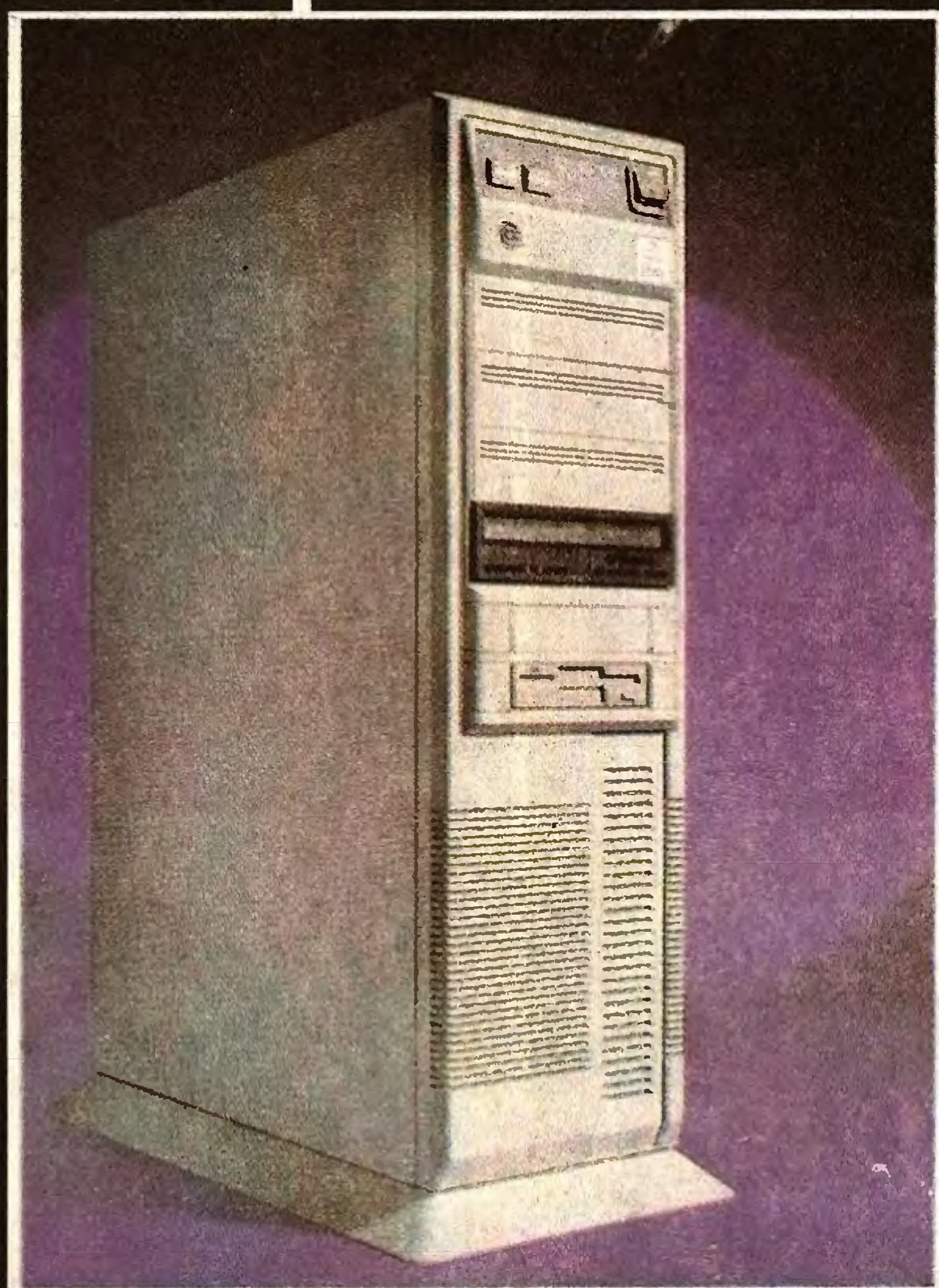
Cena komputera Tetra 030 w podstawowej wersji wynosi 9980 DM. W zestawie wyposażenia specjalnych, sprzedawanych oczywiście za dodatkową opłatą, znajduje się dysk wymienny 5,25" SyQuest o pojemności 44 MB, streamer firmy Teac 155 MB, moduł ROM (cartridge) z oprogramowaniem, różne rodzaje monitorów. Na dodatkowe opcje trzeba wydać sporo gotówki, ponieważ dysk wymienny i streamer kosztują po 1399 DM.

Użytkownik otrzymuje wraz z komputerem instrukcję obsługi i oprogramowanie. Wśród procedur systemowych znajdują się podprogramy obsługujące dodatkowe układy wejścia/wyjścia oraz rozszerzenie XBIOS. Całe urządzenie waży 21 kg.

Na podst. ST Magazin opracował

J. JARMOCH

Tetra 030 czyli Atari TT w obudowie tower



Manhattan ST

Przeciętny użytkownik Atari jest przyzwyczajony, że cała inteligentna elektronika jest umieszczona pod klawiaturą. Tymczasem większość komputerów, także profesjonalny sprzęt Atari, jest konstruowana według zupełnie innej koncepcji. Mega ST i TT mają wydzieloną jednostkę centralną, do której przyłącza się różne urządzenia, w tym również klawiaturę. Tak wyglądający sprzęt zdecydowanie dominuje w specjalistycznych sklepach komputerowych.

Ale oprócz masowo produkowanych, popularnych modeli istnieją także liczne nietypowe konstrukcje. W poprzednim numerze „Mojego Atari” opisywaliśmy komputer 190 ST 020, czyli przemysłową wersję Atari ST. Nie wyczerpuje on listy nietypowych odmian Atari. Użytkownicy wykorzystują maszyny elektroniczne w tak wielu dziedzinach nauki i techniki, że konstruktorzy standardowych urządzeń

nie zawsze mogą sprostać ich wymaganiom. Lukę tę wykorzystują mniejsze firmy, które produkują komputery w niewielkich seriach. Ich wyroby charakteryzuje się bardzo elastyczną, otwartą konstrukcją.

Taką niestandardową odmianą Atari są komputery w obudowie typu wieża (tower). Produkuje je kilka firm, ale najbardziej znana jest niemiecka firma TETRA-COMPUTER. Podstawową zaletą wieży jest duża zwartość konstrukcji, a co za tym idzie oszczędność miejsca w pomieszczeniach biurowych. Komputer taki można postawić na podłodze pod biurkiem. Ale nie jest to jedyną mocną stroną takiego rozwiązania. W obudowie tower mieści się znacznie więcej urządzeń, niż w typowym komputerze. Wieża składana jest z typowych klocków. Oprócz kilku różnych kart można w niej zamontować dodatkowy twardy dysk, napęd dysków elastycznych lub streamer. Konstrukcja systemu jest otwarta, przez co użytkownik może z łatwością dobrać potrzebne mu elementy.

MANHATTEN

Podstawowym, znanym od paru lat produktem firmy Tetra Computersysteme jest Manhattan ST. Komputer ten jest odpowiednikiem Atari Mega ST. Urządzenie jest dostarczane w wielu wersjach różniących się wyposażeniem, a od niedawna także jako zestaw do samodzielnego montażu.

Na przedniej ścianie, w górnej części obudowy wieży umieszczony jest wyłącznik sieciowy, wyłącznik z kluczykiem oraz przyciski turbo i reset. Kolorowe lampki informują użytkownika o stanie pracy komputera, włączeniu lub wyłączeniu systemu turbo, lub sygnalizują dostęp do twardego dysku. Konstruktorzy wieży Manhattan przewidzieli możliwość składania komputera z bardzo różnych elementów. Każdy z tych elementów wsuwa się od przodu do znormalizowanych gniazd. Cztery miejsca przeznaczone są na zamontowanie stacji dysków elastycznych 5.25", twardego dysku lub stacji dysków wymiennych. Oprócz tego w dwóch gniazdach można umieścić dwa napędy dysków elastycznych 3.5" o wysokości 1". Wewnątrz obudowy umieszczono zasilacz o mocy 200 W, który jest chłodzony wentylatorem

MANHATTEN ST

Komputer odpowiadający Atari Mega ST

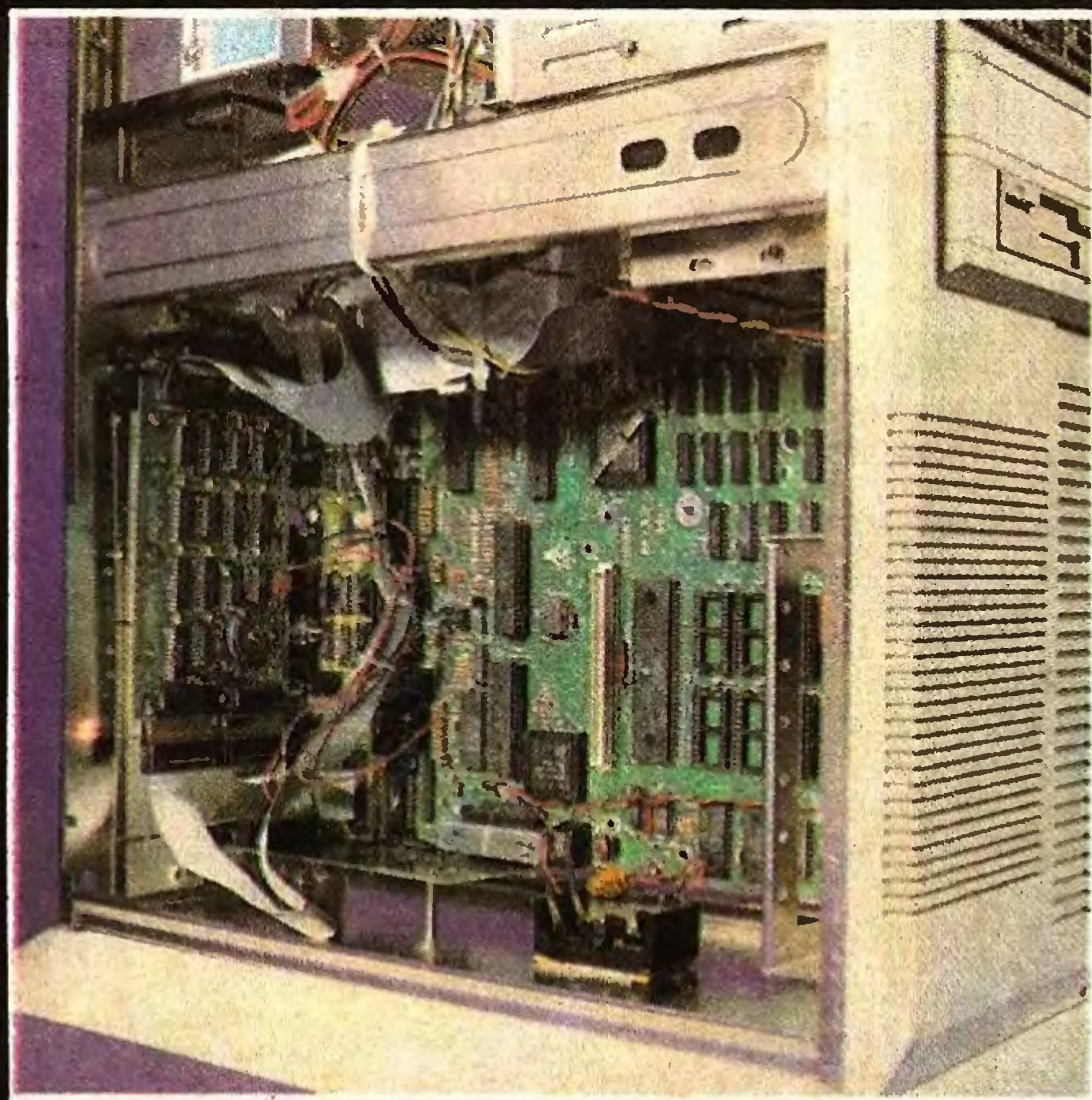
- * Procesor Motorola 68000
- * 1-4 MB RAM
- * 4 gniazda do zainstalowania napędu dysków elastycznych 5,25" (lub twardego dysku albo dysku wymiennego)
- * 2 gniazda do zainstalowania napędu dysków elastycznych 3,5"
- * Komplet standardowych złącz

KOMPUTER DLA MAJSTERKOWICZA

W samodzielnym montażu komputera pomaga 63-stronicowa instrukcja. Najtrudniejsze operacje są ilu-

port ROM. Firma Tetra wzbogaciła oryginalne rozwiązania Atari o własne usprawnienia rozszerzające możliwości Manitennu ST. Są to układy logiczne automatycznie przełączające tryb kolorowy i monochromatyczny, układ umożliwiający pracę kontrolera dysków elastycznych z większą częstotliwością, a tym samym korzystanie z dyskietek o poczwórnej gęstości. Dodatkowy sygnał w zmodyfikowanym złączu równoległym umożliwia sterowanie skanerem. Miłośników muzyki komputerowej zainteresuje wzmacniacz audio.

Otwory montażowe wszystkich płytek są wykonane bardzo dokładnie i pasują do otworów obudowy. Majsterkowicz musi samodzielnie wykonać pewną liczbę połączeń, ale kable są bardzo dobrze oznaczone kolorami i trudno jest pomylić się w ich właściwym poprowadzeniu. Końcowy efekt i poprawność montażu można sprawdzić przy pomocy specjalnego oprogramowania, które kompleksowo testuje funkcjonowanie poszczególnych



strowane zdjęciami i objaśnieniami, które pozwalają uniknąć błędów. Majsterkowicz znajdzie tu także wskazówki dotyczące wbudowania dodatkowych kart graficznych, kart przyspieszających pracę komputera, emulatorów, kart interfejsów, np. interfejsu drukarki laserowej lub interfejsu w standardzie SCSI.

Zestaw zawiera wszystkie niezbędne elementy montażowe: kable, śruby, gniazda, tulejki dystansowe, no i oczywiście płytę główną Atari Mega ST oraz dodatkowe płyty z innymi układami elektronicznymi. Komputer posiada wszystkie złącza charakterystyczne dla Mega ST, a wśród nich gniazda do przyłączenia monitora i klawiatury, interfejs MIDI, RS232, interfejs równoległy,

elementów komputera. Manhattan można kupić już za 1199 DM.

WIEŻA ATARI TT

Manhattan ST nie jest jedynym urządzeniem oferowanym przez firmę Tetra Computer. Ambitni konstruktorzy pokusili się także o opracowanie wersji Atari TT w obudowie tower. Komputer ten nazywa się Tetra 030.

Ta wieża ma nowoczesny i nieco awangardowy wygląd. W jej wnętrzu kryje się komputer TT z 8 MB pamięci RAM. Komputer zbudowano, podobnie jak TT, na bazie procesora Motorola

Dokończenie na str. 31

WIEŻA -

NIE TYLKO FIGURA SZACHOWA

Wnętrze Manitennu ze zmodyfikowaną płytą ST