

MOJE Atari 2

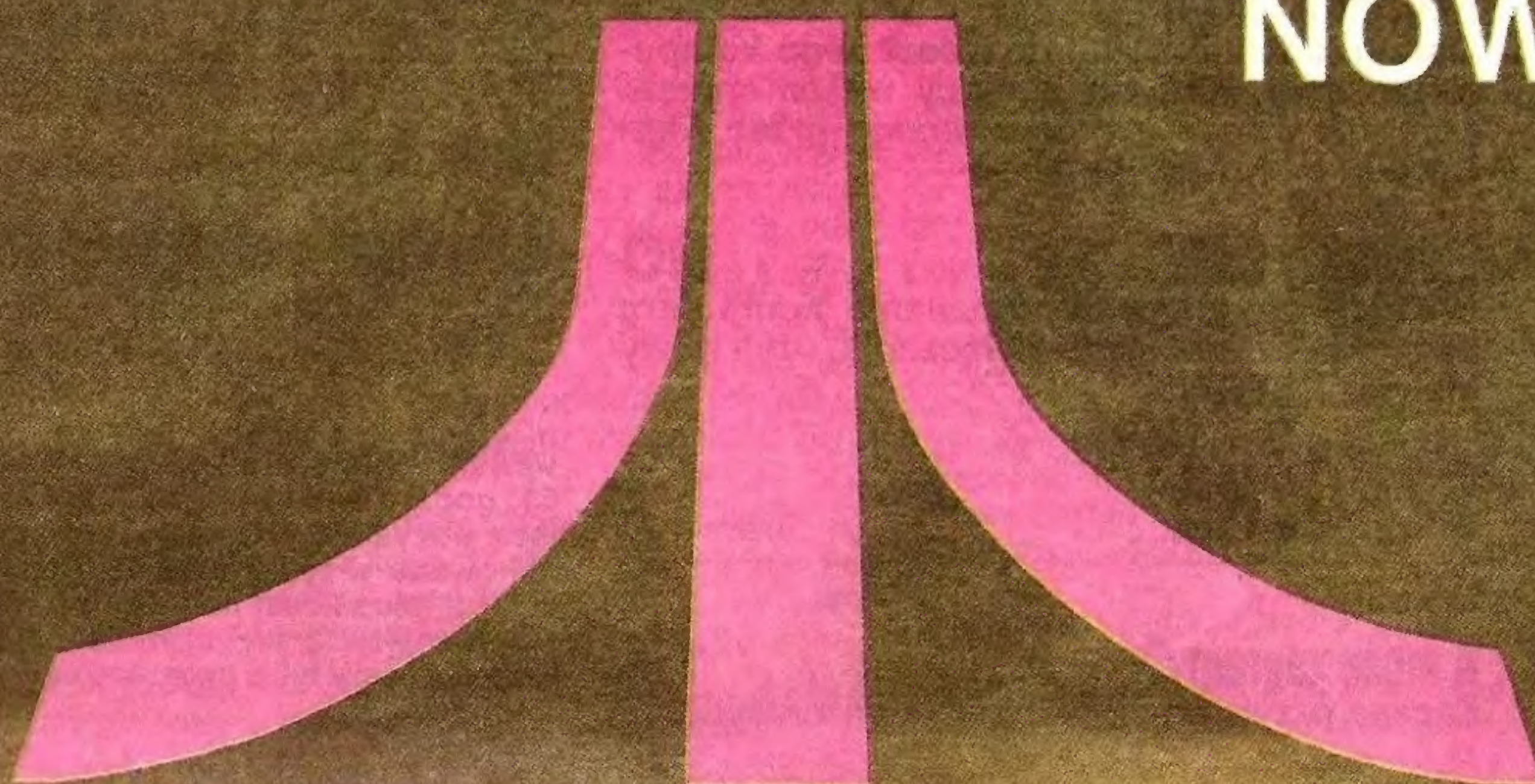
CENA 4500 zł

MAGAZYN UŻYTKOWNIKÓW KOMPUTERÓW ATARI GRUDZIEŃ 1990

MULTI
DOS
GRA
W KOŚCI
PROTOKÓŁ
MIDI

NOWY RESET

Atari
PC4



Drogi Czytelniku!

Pierwszy numer nowego pisma — „Moje Atari” — był jakby honorowym startem. Teraz czas na start ostry. Mamy wiele nowych (i starych) pomysłów dotyczących zawartości pisma. W tej sprawie liczymy też na pomoc Czytelników. Czekamy na Wasze propozycje — programy, opisy modyfikacji itd.

Planujemy poświęcenie około 1/4 objętości „Mojego Atari” komputerom serii ST. Chcielibyśmy bowiem, aby nasze pismo było czytane przez użytkowników wszystkich Atari zarówno „małych” — XL i XE, jak i „dużych” ST. Problemem jest tylko określenie takiego zakresu poruszanych tematów, który zadowoliliby naszych Czytelników.

Pierwszy z pomysłów, który doczekał się realizacji, jest powszechnie praktykowany na całym świecie. Tylko w Polsce jest to nowość.

Od bieżącego numeru wszystkie programy dla Atari XL/XE, których listingi zostały wydrukowane, będą dostępne również na dyskietkach. Niestety, ze względu na powszechnie znany niedowład poczty oraz znaczne opłaty, dyskietki te można będzie nabyć tylko bezpośrednio w redakcji. Natomiast z przyczyn technicznych nie możemy programów rozprowadzać na kasetach (przynajmniej na razie). Zainteresowane osoby muszą więc we własnym zakresie skopiować te programy na kasetę.

A propos kopiowania. Na podstawie obowiązującego prawa autorskiego drukowane w naszym piśmie programy są publicznie dostępne. Zabronione jest więc ich sprzedawanie oraz powielanie dla celów zarobkowych. Nie dotyczy to tylko opłaty za usługę, czyli skopiowanie na kasetę lub inną dyskietkę. Zabronione są także inne sposoby komercyjnego ich wykorzystania. Zastosowanie programów użytkowych lub procedur pomocniczych we własnych programach jest dozwolone tylko po podaniu źródła. Po tych kilku uwagach zapraszam do lektury.

Wojciech Zientara

MOJE **Atari**

**MAGAZYN
UŻYTKOWNIKÓW
KOMPUTERÓW ATARI**

Redaktor naczelny:
Wojciech Zientara
Opracowanie graficzne:
Wanda Roszkowska
Redaktor techniczny:
Maria Radzimińska
Zdjęcia:
Leopold Dzikowski
Wydawca:
Spółdzielnia „Bajtek”,
00-687 Warszawa, ul. Wspólna 61,
tel. 21-12-05
Druk:
Prasowe Zakłady Graficzne
w Ciechanowie
Fotoskład:
Grażyna Kurzątkowska
Montaż offsetowy:
Grażyna Ostaszewska
Korekta:
Maria Krajewska
Teresa Rutkowska
Nr. zlecenia: 71030
Nakład: 105 tys.

CO W numerze



PROGRAMOWANIE

Action! (4) 15
Kolejny odcinek opisu języka Action! poświęcony jest parametrom procedur, dyrektywom kompilatora i rozszerzonym typom danych.

Jeszcze o duszkach 10
Wykorzystanie pełnych możliwości grafiki graczy i pocisków wymaga sporego nakładu pracy. Oto program, który większość operacji przetrzuca na barki komputera.

GRY

Gra w kości 6
Popularna gra w wersji dla Atari. Można się w niej zmierzyć zarówno z komputerem, jak i z trzema kolegami.

Nowe i stare 24
Opisy gier „Rebound”, „Arctic Antics”, „Trains”, „Feud” i „221 Baker Street”.

Mercenary 21
Mapa miasta z pierwszej części gry „Mercenary” — „Escape from Targ”.

PROGRAMY UŻYTKOWE

MultiDOS 12
Program kopiujący, który przekształca programy w formacie „Multi-Boot” oraz zwykłe programy kasetowe na normalne pliki odczytywane przez DOS.

SPRZĘT

Rodzina Atari (4) 5
Ostatni odcinek cyklu poświęconego komputerom serii Atari zawierający wykaz sprzętu komputerowego produkowanego przez Atari Corp.

Sprzętowy RESET 20
Układ powodujący wyzerowanie komputera bez konieczności jego wyłączania, czyli nowy klawisz RESET.

SZKÓŁKA

A może zagramy? 14
Łącząc przyjemne z pożytecznym zaczynamy

układanie gry, na której przykładzie poznamy kolejne etapy pisania programu.

INSTRUKCJE

Soft Synth 22
Instrukcja popularnego programu służącego do układania i wykonywania prostych utworów muzycznych.

DZIAŁ ST

Protokół MIDI 30
Kompletny opis standardu przesyłania informacji poprzez złącze MIDI pozwoli na pisanie własnych programów, które będą mogły z niego korzystać.

INNE

Ankieta Atari 8
Zamiast wywiadu publikujemy wyniki ankiety ogłoszonej w drugim wydaniu specjalnym „Tylko o Atari”.

Atari PC 4 32
Pokazany na ubiegłorocznych targach CeBIT rewelacyjny „klon” IBM w wydaniu firmy Jacka Tramiela.

Adresy 20
Po firmach wydawniczych czas na adresy producentów urządzeń peryferyjnych przeznaczonych dla komputerów Atari.

8 razy Atari 28
Recenzja ośmiu czasopism zachodnich poświęconych komputerom Atari, zarówno serii XL/XE, jak i ST.

Wyścig kotów 19
Ocena prostej gry nadesłanej przez Czytelnika.

Edytor Basica 4

Wpisywanie znaków specjalnych . . . 4

Więści 3

UWAGA: W naszej redakcji (Warszawa, ul. Wspólna 61, godz. 10—15) można nabyć dyskietkę zawierającą wszystkie programy, których wydruki są opublikowane w tym numerze. Cena dyskietki wynosi 5000 zł, plus koszt nośnika lub dyskietka na wymianę. Niestety, z przyczyn technicznych nie rozpowszechniamy tych programów na kasetach magnetofonowych.



DESKTOP VIDEO

Na licznych targach odbywających się w RFN podziw zwiedzających wzbudzą niezwykle możliwości graficzne komputerowych systemów, przy pomocy których specjaliści od reklamy kreują barwne graficzne pokazy. Wymyślono już nawet nowy termin trafnie oddający istotę zastosowanego tu nowego rodzaju oprogramowania: Desktop Video.

W produkcji systemów Desktop Video opartych na Mega ST specjalizuje się firma Imagic Grafik. Opracowała ona oryginalne oprogramowanie i szereg urządzeń służących uzyskaniu obrazu i animacji spełniających profesjonalne wymagania. System umożliwia sprzężenie poprzez złącza MIDI do 256 komputerów Mega ST. Z monitorów komputerów buduje się ścianę — ogromny ekran o wymiarach kilku metrów. System synchronizuje obraz wyświetlany na monitorach. Tańczące dziewczęta lub film animowany wywierają ogromne wrażenie na widzach takiego pokazu graficznego.

W ofercie Imagic Grafik znajduje się rów-

nież Genlock-Interface, profesjonalne urządzenie przeznaczone do współpracy z ST. Umożliwia ono tworzenie kompozycji i łączenie obrazów z kamery video i obrazu generowanego przez komputer. Najprostszym przykładem możliwych do wykonania czynności jest połączenie zdjęć z kamery video z tytułami i tekstem. Jakość obrazu jest zbliżona do jakości zdjęć uzyskiwanych w tradycyjnej fotografii.

Obecnie Imagic Grafik opracowuje graficzne pokazy w oparciu o najnowszą wersję karty graficznej MGE (MAXON Graphic Expansion). Karta ta jest skonstruowana na bazie bardzo szybkiego procesora Intel 82786. Jedną z jej wersji umożliwia operowanie paletą 16.700.000 kolorów na monitorze o przekątnej 95 cm (37 cali). Nie trudno przewidzieć, że barwy tworzonego w ten sposób obrazu z powodzeniem mogą konkurować z widokami obserwowanymi w naturze.

(J.J.)

POGOŃ ZA MEGABAJTAMI

Producenci prześcigają się w opracowywaniu coraz wymyślniejszych urządzeń zwiększających możliwości Mega ST. Firma Richter Distributor z miejscowości Gevelsberg oferuje użytkownikom rozszerzenie pamięci Mega ST aż do 12 MB. Dystrybutor zapewnia przy tym, że

rozszerzenie nie wymaga stosowania dodatkowych programów zarządzających pamięcią i całe 12 MB RAM jest automatycznie rozpoznawane przez system operacyjny oraz każdy Basic. Na tak rozbudowanym Mega ST można bez przeszkód zadeklarować tablicę zawierającą nawet 1.500.000 liczb. Niestety za wygodę trzeba słono zapłacić. Pełne rozszerzenie do 12 MB kosztuje aż 3998 DM.

(J.J.)

LEGALIZACJA
PIRACKICH
KOPII



Firma Tommy Software z Berlina postanowiła od 1.07.1990 zalegalizować wszystkie pirackie kopie wyprodukowanych przez siebie programów, które będą w NRD. Producent zamierza w ten sposób dać wszystkim użytkownikom w NRD możliwość legalnego korzystania z bezprawnie skopiowanych programów. Jedynym warunkiem jest listowne zwrócenie się do firmy o bezpłatne nadanie numeru rejestracyjnego. Akcja trwa do 30.09.1990. Wnioski nadesłane w późniejszym terminie nie będą rozpatrywane. W NRD istnieje pokaźny rynek polskiego oprogramowania. Tommy Software podejmując taką decyzję strategiczną na pewno nie kierowała się sentymentami. Firma zamierza w ten sposób zabezpieczyć swoje interesy. Pomysłodawca liczy na przyciągnięcie wielu potencjalnych klientów, którzy zetknęli się już z jego wyrobami.

(J.J.)



MEGA ST W SALONIE BMW

Na ostatniej Międzynarodowej Wystawie Samochodowej we Frankfurcie firma BMW zatrudniła do pracy w swoich stoiskach Mega ST, powierzając mu odpowiedzialne zadanie informowania klientów o swoich wyrobach. Obok każdego wystawianego samochodu znajdowała się konsola. Najważniejszym jej elementem było oczywiście Mega ST. W konsoli zainstalowano również ekran ciekłokrystaliczny o wysokiej rozdzielczości. Z elektronicznego informatora zadowolili byli i zwiedzający, i wystawcy. Tym pierwszym imponowała cierpliwość z jaką komputer odpowiadał na najciekawsze pytania. Drudzy mogli dzięki ST odetchnąć nieco więcej niż zwykle. Producentem konsoli jest firma Imagic Grafik.

(J.J.)

EDYTOR

BASIC-a

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 przepisaną można sprawdzić samym „Edytorem” w opisany niżej sposób.

Przystępując do wpisywania dowolnego programu z naszego pisma trzeba najpierw wczytać i uruchomić „Edytora Basica”. Następnie należy przepisywać kolejne wiersze programu. Po wpisaniu każdego wiersza i naciśnięciu klawisza RETURN wiersz ten pojawia się w dowolnej 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 ponownie 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 "Bajtka"
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$(D,D))):NEXT D
SR 32160 KOD=INT(ODP/676)
BX 32170 KOD=ODP-KOD*676
BM 32180 KODS=INT(KOD/26)
VF 32190 KODM=KOD-(KODS*26)+193
NF 32200 KODS=KODS+193
UW 32210 POSITION 0,16:? CHR$(KODS);CHR$(
    KODM)
HN 32220 POSITION 2,13:? "Jezeli kod sie
    nie zgadza, naciśnij RETURN i popr
    aw linie.":GOTO 32050
```

WPISYWANIE ZNAKÓW SPECJALNYCH

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 znajdują się obok siebie są drukowane bez odstępów (jak na ekranie monitora), lecz między poszczególnymi wierszami pozostaje niewielki odstęp (inaczej niż na ekranie). Poniżej pokazany jest wygląd wszystkich znaków uwidocznionych na klawiaturze.

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
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 tabelach „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 Atari). 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 te są pokazane poniżej.

SPECJALNE

< ▤ CTRL F
= ▤ CTRL G
+ ▤ CTRL N
+ ▤ CTRL R
+ ▤ CTRL S

NORMALNE

< ▤ znak /
= ▤ SHIFT +
+ ▤ SHIFT -
+ ▤ znak -
+ ▤ znak +

NORMALNIE

↑	CTRL ,	+	CTRL S
↑	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 S
↑	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	↑	

rodzina

Atari (4)

Na zakończenie cyklu o ośmiobitowym sprzęcie komputerowym firmy Atari prezentujemy krótki przewodnik. Umożliwia on szybkie rozpoznanie rodzaju i przeznaczenia urządzenia, według jego symbolu.

Rodzina Atari składa się z trzech podstawowych serii sprzętu, które oznaczane są według symboli komputerów, będących podstawą serii. W ten sposób można więc wyróżnić serie **400/800**, **XL** i **XE**. Urządzenia należące do poszczególnych serii mają numery większe od 400 lub 800 dla komputerów 400/800, większe od 1000 dla XL i rozpoczynające się od litery X dla serii XE. Oczywiście, podział ten jest czysto teoretyczny i dotyczy tylko wyglądu zewnętrznego sprzętu. Niemal wszystkie urządzenia Atari pasują bowiem do wszystkich ośmiobitowych komputerów tej firmy. Dzięki odpowiedniej polityce produkcyjnej wyjątki od tej reguły są sporadyczne. Nie dotyczy to tylko serii ST, która jest rodziną komputerów następnej generacji wyposażonych w procesor 16-bitowy. O sprzęcie z tej rodziny napiszę w przyszłości.

W seriach 400/800 i XL urządzenia z liczbą 20 są drukarkami, zaś z liczbą 30 modemami. Nieco mniejsza konsekwencja panuje w przypadku pozostałych urządzeń. Początkowo dla pamięci masowych (magnetofonów i stacji dysków) zarezerwowane były liczby z 10, a dla interfejsów z 50. Później liczba 10 pozostała tylko dla magnetofonów, zaś 50 przeznaczono dla stacji dysków, gdyż firma Atari zaprzestała produkcji interfejsów pozostawiając to innym, mniejszym przedsiębiorstwom. Po rozpoczęciu produkcji serii XE i ST przyjęto całkowicie odmienny system oznaczeń: z literą X dla XE i z literą S dla ST. Dalsze litery nazwy każdego urządzenia określają jego rodzaj: C — magnetofon (dla XE) lub monitor kolorowy (dla ST), F — stacja dysków elastycznych, H — twardy dysk, M — modem (XE) lub monitor monochromatyczny (ST) oraz drukarki: MM — mozaikowe, DM — znakowe i LM — laserowe.

Seria 400/800

- 400 — komputer 16 KB RAM
- 410 — magnetofon
- 800 — komputer 48 KB RAM
- 810 — stacja dysków (jednostronna, pojedyncza gęstość)
- 815 — stacja dysków (jednostronna, podwójna gęstość)
- 820 — 40 kolumnowa drukarka-ploter
- 825 — 80 kolumnowa drukarka mozaikowa
- 830 — modem akustyczny 300 bodów
- 835 — modem akustyczny 300 bodów
- 850 — interfejs (4 RS232 i Centronics)

Seria XL

- 600XL — komputer 16 KB RAM
- 800XL — komputer 64 KB RAM
- 1200XL — komputer 64 KB RAM

- 1400XL — komputer 64 KB RAM (nie produkowany)
- 1450XLD — komputer 64 KB RAM z wbudowaną stacją dysków
- 1010 — magnetofon
- 1020 — 40-kolumnowa drukarka-ploter
- 1025 — 80-kolumnowa drukarka mozaikowa
- 1027 — 80-kolumnowa drukarka znakowa
- 1029 — 80-kolumnowa drukarka mozaikowa
- 1030 — modem galwaniczny 300 bodów
- 1050 — stacja dysków (jednostronna, pojedyncza i rozszerzona gęstość)
- 1064 — moduł 64 KB RAM do 600XL

Seria XE

- 65XE — komputer 64 KB RAM
- 130XE — komputer 128 KB RAM
- 260XE — komputer 256 KB RAM (nie produkowany)
- 800XE — komputer 800XL w obudowie XE
- XEGS — komputer 65XE w zmienionej obudowie (*XE Game System*)
- XC11 — magnetofon
- XC12 — magnetofon
- XEP80 — interfejs 80-kolumnowego monitora
- XF35 — stacja dysków 3,5 cala (nie produkowana)
- XF551 — stacja dysków (dwustronna; pojedyncza, rozszerzona i podwójna gęstość)
- XM301 — modem galwaniczny 300 bodów
- XDM121 — 80-kolumnowa drukarka znakowa
- XMM801 — 80-kolumnowa drukarka mozaikowa

Seria ST

- 260ST — komputer 256 KB RAM
- 260ST + — komputer 512 KB RAM

- 520ST — komputer 512 KB RAM, dodatkowe litery oznaczają:
 - F — wbudowana stacja dysków,
 - M — wbudowany modulator TV
- 520ST + — komputer 1024 KB RAM
- 1040ST — komputer 1024 KB RAM (litery dodatkowe jak 520ST)
- MEGA ST2 — komputer 2 MB (2048 KB) RAM
- MEGA ST4 — komputer 4 MB (4096 KB) RAM
- SC1224 — monitor kolorowy
- SF314 — dwustronna stacja dysków (720 KB)
- SF354 — jednostronna stacja dysków (360 KB)
- SH204 — twardy dysk 20 MB
- SH205 — twardy dysk 20 MB
- SH317 — twardy dysk 10 MB
- SLM804 — drukarka laserowa
- SM124 — monitor monochromatyczny 12 cali
- SM125 — monitor monochromatyczny 12 cali
- SMM804 — 80-kolumnowa drukarka mozaikowa
- SW212 — modem galwaniczny 300/1200 bodów
- MEGAFILE 30 — twardy dysk 30 MB
- MEGAFILE 60 — twardy dysk 60 MB

Wojciech Zientara





GRA W KOŚCI

Przedstawiamy komputerową wersję popularnej gry w kości. Może w nią grać jednocześnie od jednego do czterech graczy. Wszystkie niezbędne czynności od rzucania kostek, aż po prowadzenie zapisu, są wykonywane przez komputer.

Zamieszczony obok program w Turbo Basicu XL należy uważnie przepisać. Ze względu na dużą liczbę znaków sterujących i graficznych konieczne jest wykorzystanie „Edytora Basica” w celu uniknięcia błędów. Litery znajdujące się przed numerami wierszy są kodami kontrolnymi i nie należy ich przepisywać. Przed uruchomieniem programu trzeba go jeszcze zapisać na kasecie lub dyskietce.

Obsługa programu jest bardzo prosta — wszystkie funkcje są opisane na ekranie. Zasady gry są takie same jak stosowane przy grze normalnymi kośćmi. Po zakończeniu rozgrywki program rozdziela odpowiednio premie i punkty karne oraz zapamiętuje najlepszy wynik.

Leszek Stróżowski

```

VW 10 REM GRA W KOŚCI
QC 20 REM LESZEK STROZOWSKI
MC 30 REM COPYRIGHT (C) BAJTEK
BB 40 REM
LZ 60 POKE 106,182:GRAPHICS 0:POKE 752,1:
POKE 82,0:POKE 566,158:POKE 731,1
HQ 70 EXEC POL:POKE 756,180
OJ 75 DIM G1$(3),G2$(3),G3$(3),G4$(3),S$(
3),K(4),N(60):P=-20:G1$(2,2)="" :G2$(2
,2)="" :G3$(2,2)="" :G4$(2,2)=""
EL 100 EXEC PLAN5ZA
QP 110 FOR X=1 TO 60:W(X)=-20:NEXT X
QG 115 S$1=0:S$2=0:S$3=0:S$4=0:SP1=0:SP2=
0:SP3=0:SP4=0
RG 120 EXEC TABELA
JQ 125 EXEC ILGRA
UQ 130 ON IG GOTO 140,150,170
PZ 140 FOR Q=1 TO 15:EXEC RZUCA1:PT=0:EKE
C RZUTY:EXEC COZAP:EXEC RZUCA2:PT=15:E
XEC RZUTY:EXEC COZAP:NEXT Q:G0R KONIEC
150 FOR Q=1 TO 15:EXEC RZUCA1:PT=0:EKE
C RZUTY:EXEC COZAP:EXEC RZUCA2:PT=15:E
XEC RZUTY:EXEC COZAP
EN 180 EXEC RZUCA3:PT=30:EXEC RZUTY:EXEC
COZAP:EXEC RZUCA4:PT=45:EXEC RZUTY:EKE
C COZAP:NEXT Q:G0R KONIEC
MS 185 PROC ZMAZ
IH 190 POSITION 1,21:?"[X]":
WQ 195 ENDPROC
AF 400 PROC PORZADEK
CE 410 FOR JP=1 TO 4:F=0
XF 420 FOR IP=0 TO 4-JP
MC 430 IF K(IP)<K(IP+1) THEN 460
FF 440 F=1
QW 450 TAB=K(IP+1):K(IP+1)=K(IP):K(IP)=TA
B
OC 460 NEXT IP
ZI 470 IF F=0 THEN 490
OQ 480 NEXT JP
WE 490 ENDPROC
FU 500 PROC DL
GK 510 S$=STR$(S)
TW 520 L=LEN(S$)-1
VT 530 ENDPROC
RE 550 PROC ZR
TN 560 IF Y=1 OR W=1 THEN S=S*2
WB 570 ENDPROC
MG 660 PROC TAB
NB 665 SETCOLOR 1,0,12:SETCOLOR 2,8,2:SET
COLOR 4,8,2
JE 670 MOVE 45632,47500,600:CLS :MOVE 466
80,45632,800:EXEC 503:POSITION 1,22:?"
[123] POWRÓT":
LB 675 POSITION 1,23:?"[X] ZAPISANIE WYNIK
U":
PT 680 GET KEY:IF KEY<>127 AND KEY<>90 TH
EN 690
IR 682 IF KEY=90 THEN 698
XH 684 IF KEY=127 THEN 698
DD 690 CLS :SETCOLOR 1,0,0:SETCOLOR 2,11,
6:SETCOLOR 4,11,6:MOVE 47500,45632,600
:EXEC 503:GOSUB 2250
WX 695 ENDPROC
WB 698 POP :GOTO 2150
IL 700 PROC KTOREKOSCI
TN 702 AQ=0:BQ=0:CQ=0:DQ=0:EQ=0:PA=1
DT 705 W=0:EXEC ZMAZ:POSITION 1,22:?"Kte
RYMI KOBCMI CHCESZ RZUCAC ?":POSITION
1,23
WJ 707 ? "1 2 3 4 5 6" - PO WYBORZE [52]
CA":EXEC 501
UX 710 GET KEY:IF KEY<49 AND KEY>53 AND K
EY<>32 AND KEY<>87 THEN 710
CX 720 IF KEY=32 THEN 766
KM 725 IF KEY=49 THEN GOSUB 767:GOSUB 770
NG 730 IF KEY=50 THEN GOSUB 767:GOSUB 775

```

```

IL 735 IF KEY=51 THEN GOSUB 767:GOSUB 780
PT 740 IF KEY=52 THEN GOSUB 767:GOSUB 785
KV 745 IF KEY=53 THEN GOSUB 767:GOSUB 790
CW 750 IF KEY=87 THEN 760
PE 755 GOTO 710
XD 760 GOSUB 767:GOSUB 770:PAUSE 10:GOSUB
775:PAUSE 10:GOSUB 780:PAUSE 10:GOSUB
785:PAUSE 10:GOSUB 790:W=1
HQ 765 ENDPROC
UR 766 GOSUB 767:SOUND 0,70,10,10:PAUSE 5
:DOSOUND :EXEC ZMAZ:PAUSE 10:ENDPROC
DH 767 SOUND 0,30,10,10:PAUSE 5:DOSOUND :R
ETURN
EH 770 POSITION 1,10:EXEC GUMKA:EXEC A:AQ
=1:RETURN
LY 775 POSITION 9,10:EXEC GUMKA:EXEC B:BQ
=1:RETURN
WH 780 POSITION 17,10:EXEC GUMKA:EXEC C:C
Q=1:RETURN
ZC 785 POSITION 25,10:EXEC GUMKA:EXEC D:D
Q=1:RETURN
AU 790 POSITION 33,10:EXEC GUMKA:EXEC E:E
Q=1:RETURN
PU 792 PROC GUMKA
KC 794 ? " "
WZ 796 ENDPROC
YN 800 PROC WYDRUKKOSCI
SQ 810 IF PA THEN POSITION 3,7:?"1
2
3
4
5":PAUSE 5
ZU 815 IF AQ
LT 817 IF PA THEN PAUSE 5:EXEC DZWIEM
YD 820 POSITION 1,10:ON K(0) GOSUB 890,90
0,910,920,930,940
JI 825 ENDIF
AL 827 IF BQ
LY 828 IF PA THEN PAUSE 5:EXEC DZWIEM
DL 830 POSITION 9,10:ON K(1) GOSUB 890,90
0,910,920,930,940
JK 835 ENDIF
AU 837 IF CQ
MA 838 IF PA THEN PAUSE 5:EXEC DZWIEM
MK 840 POSITION 17,10:ON K(2) GOSUB 890,9
00,910,920,930,940
JM 845 ENDIF
BF 847 IF DQ
MC 848 IF PA THEN PAUSE 5:EXEC DZWIEM
MU 850 POSITION 25,10:ON K(3) GOSUB 890,9
00,910,920,930,940
JO 855 ENDIF
BP 857 IF EQ
ME 858 IF PA THEN PAUSE 5:EXEC DZWIEM
NG 860 POSITION 33,10:ON K(4) GOSUB 890,9
00,910,920,930,940
JQ 865 ENDIF
WE 870 ENDPROC
CN 890 ? " "
NC 900 ? " "
YK 910 ? " "
JS 920 ? " "
VA 930 ? " "
GI 940 ? " "
ZA 950 PROC A
LF 952 K(0)=INT(RND*6+1)
WQ 955 ENPROC
ZM 960 PROC B
LO 962 K(1)=INT(RND*6+1)
NS 965 ENDPROC
ZY 970 PROC C
LX 972 K(2)=INT(RND*6+1)
NU 975 ENDPROC
AK 980 PROC D
MG 982 K(3)=INT(RND*6+1)
NH 985 ENDPROC
AW 990 PROC E
MP 992 K(4)=INT(RND*6+1)

```



```

MY 995 ENDPROC
JM 1000 PROC RZUCA1
PN 1010 GOSUB 1200:TEXT 28,26,G1$
OR 1030 GOSUB 1195: G1$
YO 1040 ENDPROC
KR 1050 PROC RZUCA2
RF 1060 GOSUB 1200:TEXT 28,26,G2$
PZ 1080 GOSUB 1195: G2$
ZD 1090 ENDPROC
KU 1100 PROC RZUCA3
RV 1110 GOSUB 1200:TEXT 28,26,G3$
QG 1130 GOSUB 1195: G3$
YQ 1140 ENDPROC
LZ 1150 PROC RZUCA4
TN 1160 GOSUB 1200:TEXT 28,26,G4$
RP 1180 GOSUB 1195: G4$
ZF 1190 ENDPROC
VY 1195 FOR X=1 TO 3:SETCOLOR 0,5,6:SOUND
0,15,12,10:PAUSE 20:SETCOLOR 0,0,0:D5
OUND:PAUSE 20:NEXT X
FC 1196 GRAPHICS 0:POKE 752,1:POKE 756,18
8
JK 1197 SETCOLOR 1,0,0:SETCOLOR 2,11,6:SE
TCOLOR 4,11,6:POSITION 1,0: "RZUCA "
:RETURN
JG 1200 GRAPHICS 20:COLOR 1:SETCOLOR 4,0,
0:SETCOLOR 0,0,0:TEXT 20,10,"RZUCA":RE
TURN
CJ 2000 PROC RZUTY
MX 2005 POSITION 32,0: "RZUT 1":
JD 2010 EXEC A:EXEC B:EXEC C:EXEC D:EXEC
E
YQ 2015 AQ=1:BQ=1:CQ=1:DQ=1:EQ=1:PA=1
CN 2020 FOR Y=1 TO 2
XN 2030 EXEC HYDRUKK05CI
XN 2040 GOSUB 2250:EXEC 501
XG 2050 GET KEY:IF KEY<78 AND KEY<84 AN
D KEY<27 AND KEY<127 THEN 2050
KV 2060 IF KEY=27 THEN 2210
UL 2070 IF KEY=127 THEN EXEC TAB
NG 2080 IF KEY=78 THEN 2110
LR 2090 IF KEY=84 THEN 2200
PP 2100 GOTO 2050
DP 2110 EXEC KTOREK05CI
WQ 2115 POSITION 37,0: Y+1
LS 2120 NEXT Y
EQ 2130 EXEC HYDRUKK05CI:EXEC ZMAZ:POSITI
ON 1,23: "TO BYD OSTATNI RZUT":
KV 2132 SOUND 0,100,10,5:PAUSE 10:SOUND 0
,200,10,15:PAUSE 10:SOUND 0,150,10,10:
PAUSE 10:D5OUND:PAUSE 100
BA 2135 SETCOLOR 1,0,12:SETCOLOR 2,8,2:SE
TCOLOR 4,8,2
CU 2140 MOVE 46680,45632,800
PW 2150 EXEC PORZADEK
MA 2160 GOSUB 2300:EXEC 501
ZG 2190 ENDPROC
MY 2200 POP:GOTO 2135
IU 2210 EXEC PORZADEK:AQ=1:BQ=1:CQ=1:DQ=1
:EQ=1:PA=0:EXEC 504:EXEC HYDRUKK05CI:G
OTO 2050
DR 2250 EXEC ZMAZ:POSITION 1,23: "CZY ZA
PISAC WYNIK ? [Y/N]":POSITION 1,21: "3
59 UPORZADKOWANIE K06CI":
KA 2255 POSITION 1,22: "TABELA WYNIK
04W":
AH 2260 RETURN
LD 2300 EXEC ZMAZ:POSITION 1,21: "
KC 2310 POSITION 1,22: "K(0):" K(0):K
(1):K(2):K(3):K(4):K(5):K(6):K(7):K(8):K
(9):K(10):K(11):K(12):K(13):K(14):K(15):K
(16):K(17):K(18):K(19):K(20):K(21):K(22):K
(23):K(24):K(25):K(26):K(27):K(28):K(29):K
(30):K(31):K(32):K(33):K(34):K(35):K(36):K
(37):K(38):K(39):K(40):K(41):K(42):K(43):K
(44):K(45):K(46):K(47):K(48):K(49):K(50):K
(51):K(52):K(53):K(54):K(55):K(56):K(57):K
(58):K(59):K(60):K(61):K(62):K(63):K(64):K
(65):K(66):K(67):K(68):K(69):K(70):K(71):K
(72):K(73):K(74):K(75):K(76):K(77):K(78):K
(79):K(80):K(81):K(82):K(83):K(84):K(85):K
(86):K(87):K(88):K(89):K(90):K(91):K(92):K
(93):K(94):K(95):K(96):K(97):K(98):K(99):K
(100):K(101):K(102):K(103):K(104):K(105):K
(106):K(107):K(108):K(109):K(110):K(111):K
(112):K(113):K(114):K(115):K(116):K(117):K
(118):K(119):K(120):K(121):K(122):K(123):K
(124):K(125):K(126):K(127):K(128):K(129):K
(130):K(131):K(132):K(133):K(134):K(135):K
(136):K(137):K(138):K(139):K(140):K(141):K
(142):K(143):K(144):K(145):K(146):K(147):K
(148):K(149):K(150):K(151):K(152):K(153):K
(154):K(155):K(156):K(157):K(158):K(159):K
(160):K(161):K(162):K(163):K(164):K(165):K
(166):K(167):K(168):K(169):K(170):K(171):K
(172):K(173):K(174):K(175):K(176):K(177):K
(178):K(179):K(180):K(181):K(182):K(183):K
(184):K(185):K(186):K(187):K(188):K(189):K
(190):K(191):K(192):K(193):K(194):K(195):K
(196):K(197):K(198):K(199):K(200):K(201):K
(202):K(203):K(204):K(205):K(206):K(207):K
(208):K(209):K(210):K(211):K(212):K(213):K
(214):K(215):K(216):K(217):K(218):K(219):K
(220):K(221):K(222):K(223):K(224):K(225):K
(226):K(227):K(228):K(229):K(230):K(231):K
(232):K(233):K(234):K(235):K(236):K(237):K
(238):K(239):K(240):K(241):K(242):K(243):K
(244):K(245):K(246):K(247):K(248):K(249):K
(250):K(251):K(252):K(253):K(254):K(255):K
(256):K(257):K(258):K(259):K(260):K(261):K
(262):K(263):K(264):K(265):K(266):K(267):K
(268):K(269):K(270):K(271):K(272):K(273):K
(274):K(275):K(276):K(277):K(278):K(279):K
(280):K(281):K(282):K(283):K(284):K(285):K
(286):K(287):K(288):K(289):K(290):K(291):K
(292):K(293):K(294):K(295):K(296):K(297):K
(298):K(299):K(300):K(301):K(302):K(303):K
(304):K(305):K(306):K(307):K(308):K(309):K
(310):K(311):K(312):K(313):K(314):K(315):K
(316):K(317):K(318):K(319):K(320):K(321):K
(322):K(323):K(324):K(325):K(326):K(327):K
(328):K(329):K(330):K(331):K(332):K(333):K
(334):K(335):K(336):K(337):K(338):K(339):K
(340):K(341):K(342):K(343):K(344):K(345):K
(346):K(347):K(348):K(349):K(350):K(351):K
(352):K(353):K(354):K(355):K(356):K(357):K
(358):K(359):K(360):K(361):K(362):K(363):K
(364):K(365):K(366):K(367):K(368):K(369):K
(370):K(371):K(372):K(373):K(374):K(375):K
(376):K(377):K(378):K(379):K(380):K(381):K
(382):K(383):K(384):K(385):K(386):K(387):K
(388):K(389):K(390):K(391):K(392):K(393):K
(394):K(395):K(396):K(397):K(398):K(399):K
(400):K(401):K(402):K(403):K(404):K(405):K
(406):K(407):K(408):K(409):K(410):K(411):K
(412):K(413):K(414):K(415):K(416):K(417):K
(418):K(419):K(420):K(421):K(422):K(423):K
(424):K(425):K(426):K(427):K(428):K(429):K
(430):K(431):K(432):K(433):K(434):K(435):K
(436):K(437):K(438):K(439):K(440):K(441):K
(442):K(443):K(444):K(445):K(446):K(447):K
(448):K(449):K(450):K(451):K(452):K(453):K
(454):K(455):K(456):K(457):K(458):K(459):K
(460):K(461):K(462):K(463):K(464):K(465):K
(466):K(467):K(468):K(469):K(470):K(471):K
(472):K(473):K(474):K(475):K(476):K(477):K
(478):K(479):K(480):K(481):K(482):K(483):K
(484):K(485):K(486):K(487):K(488):K(489):K
(490):K(491):K(492):K(493):K(494):K(495):K
(496):K(497):K(498):K(499):K(500):K(501):K
(502):K(503):K(504):K(505):K(506):K(507):K
(508):K(509):K(510):K(511):K(512):K(513):K
(514):K(515):K(516):K(517):K(518):K(519):K
(520):K(521):K(522):K(523):K(524):K(525):K
(526):K(527):K(528):K(529):K(530):K(531):K
(532):K(533):K(534):K(535):K(536):K(537):K
(538):K(539):K(540):K(541):K(542):K(543):K
(544):K(545):K(546):K(547):K(548):K(549):K
(550):K(551):K(552):K(553):K(554):K(555):K
(556):K(557):K(558):K(559):K(560):K(561):K
(562):K(563):K(564):K(565):K(566):K(567):K
(568):K(569):K(570):K(571):K(572):K(573):K
(574):K(575):K(576):K(577):K(578):K(579):K
(580):K(581):K(582):K(583):K(584):K(585):K
(586):K(587):K(588):K(589):K(590):K(591):K
(592):K(593):K(594):K(595):K(596):K(597):K
(598):K(599):K(600):K(601):K(602):K(603):K
(604):K(605):K(606):K(607):K(608):K(609):K
(610):K(611):K(612):K(613):K(614):K(615):K
(616):K(617):K(618):K(619):K(620):K(621):K
(622):K(623):K(624):K(625):K(626):K(627):K
(628):K(629):K(630):K(631):K(632):K(633):K
(634):K(635):K(636):K(637):K(638):K(639):K
(640):K(641):K(642):K(643):K(644):K(645):K
(646):K(647):K(648):K(649):K(650):K(651):K
(652):K(653):K(654):K(655):K(656):K(657):K
(658):K(659):K(660):K(661):K(662):K(663):K
(664):K(665):K(666):K(667):K(668):K(669):K
(670):K(671):K(672):K(673):K(674):K(675):K
(676):K(677):K(678):K(679):K(680):K(681):K
(682):K(683):K(684):K(685):K(686):K(687):K
(688):K(689):K(690):K(691):K(692):K(693):K
(694):K(695):K(696):K(697):K(698):K(699):K
(700):K(701):K(702):K(703):K(704):K(705):K
(706):K(707):K(708):K(709):K(710):K(711):K
(712):K(713):K(714):K(715):K(716):K(717):K
(718):K(719):K(720):K(721):K(722):K(723):K
(724):K(725):K(726):K(727):K(728):K(729):K
(730):K(731):K(732):K(733):K(734):K(735):K
(736):K(737):K(738):K(739):K(740):K(741):K
(742):K(743):K(744):K(745):K(746):K(747):K
(748):K(749):K(750):K(751):K(752):K(753):K
(754):K(755):K(756):K(757):K(758):K(759):K
(760):K(761):K(762):K(763):K(764):K(765):K
(766):K(767):K(768):K(769):K(770):K(771):K
(772):K(773):K(774):K(775):K(776):K(777):K
(778):K(779):K(780):K(781):K(782):K(783):K
(784):K(785):K(786):K(787):K(788):K(789):K
(790):K(791):K(792):K(793):K(794):K(795):K
(796):K(797):K(798):K(799):K(800):K(801):K
(802):K(803):K(804):K(805):K(806):K(807):K
(808):K(809):K(810):K(811):K(812):K(813):K
(814):K(815):K(816):K(817):K(818):K(819):K
(820):K(821):K(822):K(823):K(824):K(825):K
(826):K(827):K(828):K(829):K(830):K(831):K
(832):K(833):K(834):K(835):K(836):K(837):K
(838):K(839):K(840):K(841):K(842):K(843):K
(844):K(845):K(846):K(847):K(848):K(849):K
(850):K(851):K(852):K(853):K(854):K(855):K
(856):K(857):K(858):K(859):K(860):K(861):K
(862):K(863):K(864):K(865):K(866):K(867):K
(868):K(869):K(870):K(871):K(872):K(873):K
(874):K(875):K(876):K(877):K(878):K(879):K
(880):K(881):K(882):K(883):K(884):K(885):K
(886):K(887):K(888):K(889):K(890):K(891):K
(892):K(893):K(894):K(895):K(896):K(897):K
(898):K(899):K(900):K(901):K(902):K(903):K
(904):K(905):K(906):K(907):K(908):K(909):K
(910):K(911):K(912):K(913):K(914):K(915):K
(916):K(917):K(918):K(919):K(920):K(921):K
(922):K(923):K(924):K(925):K(926):K(927):K
(928):K(929):K(930):K(931):K(932):K(933):K
(934):K(935):K(936):K(937):K(938):K(939):K
(940):K(941):K(942):K(943):K(944):K(945):K
(946):K(947):K(948):K(949):K(950):K(951):K
(952):K(953):K(954):K(955):K(956):K(957):K
(958):K(959):K(960):K(961):K(962):K(963):K
(964):K(965):K(966):K(967):K(968):K(969):K
(970):K(971):K(972):K(973):K(974):K(975):K
(976):K(977):K(978):K(979):K(980):K(981):K
(982):K(983):K(984):K(985):K(986):K(987):K
(988):K(989):K(990):K(991):K(992):K(993):K
(994):K(995):K(996):K(997):K(998):K(999):K
(1000):K(1001):K(1002):K(1003):K(1004):K
(1005):K(1006):K(1007):K(1008):K(1009):K
(1010):K(1011):K(1012):K(1013):K(1014):K
(1015):K(1016):K(1017):K(1018):K(1019):K
(1020):K(1021):K(1022):K(1023):K(1024):K
(1025):K(1026):K(1027):K(1028):K(1029):K
(1030):K(1031):K(1032):K(1033):K(1034):K
(1035):K(1036):K(1037):K(1038):K(1039):K
(1040):K(1041):K(1042):K(1043):K(1044):K
(1045):K(1046):K(1047):K(1048):K(1049):K
(1050):K(1051):K(1052):K(1053):K(1054):K
(1055):K(1056):K(1057):K(1058):K(1059):K
(1060):K(1061):K(1062):K(1063):K(1064):K
(1065):K(1066):K(1067):K(1068):K(1069):K
(1070):K(1071):K(1072):K(1073):K(1074):K
(1075):K(1076):K(1077):K(1078):K(1079):K
(1080):K(1081):K(1082):K(1083):K(1084):K
(1085):K(1086):K(1087):K(1088):K(1089):K
(1090):K(1091):K(1092):K(1093):K(1094):K
(1095):K(1096):K(1097):K(1098):K(1099):K
(1100):K(1101):K(1102):K(1103):K(1104):K
(1105):K(1106):K(1107):K(1108):K(1109):K
(1110):K(1111):K(1112):K(1113):K(1114):K
(1115):K(1116):K(1117):K(1118):K(1119):K
(1120):K(1121):K(1122):K(1123):K(1124):K
(1125):K(1126):K(1127):K(1128):K(1129):K
(1130):K(1131):K(1132):K(1133):K(1134):K
(1135):K(1136):K(1137):K(1138):K(1139):K
(1140):K(1141):K(1142):K(1143):K(1144):K
(1145):K(1146):K(1147):K(1148):K(1149):K
(1150):K(1151):K(1152):K(1153):K(1154):K
(1155):K(1156):K(1157):K(1158):K(1159):K
(1160):K(1161):K(1162):K(1163):K(1164):K
(1165):K(1166):K(1167):K(1168):K(1169):K
(1170):K(1171):K(1172):K(1173):K(1174):K
(1175):K(1176):K(1177):K(1178):K(1179):K
(1180):K(1181):K(1182):K(1183):K(1184):K
(1185):K(1186):K(1187):K(1188):K(1189):K
(1190):K(1191):K(1192):K(1193):K(1194):K
(1195):K(1196):K(1197):K(1198):K(1199):K
(1200):K(1201):K(1202):K(1203):K(1204):K
(1205):K(1206):K(1207):K(1208):K(1209):K
(1210):K(1211):K(1212):K(1213):K(1214):K
(1215):K(1216):K(1217):K(1218):K(1219):K
(1220):K(1221):K(1222):K(1223):K(1224):K
(1225):K(1226):K(1227):K(1228):K(1229):K
(1230):K(1231):K(1232):K(1233):K(1234):K
(1235):K(1236):K(1237):K(1238):K(1239):K
(1240):K(1241):K(1242):K(1243):K(1244):K
(1245):K(1246):K(1247):K(1248):K(1249):K
(1250):K(1251):K(1252):K(1253):K(1254):K
(1255):K(1256):K(1257):K(1258):K(1259):K
(1260):K(1261):K(1262):K(1263):K(1264):K
(1265):K(1266):K(1267):K(1268):K(1269):K
(1270):K(1271):K(1272):K(1273):K(1274):K
(1275):K(1276):K(1277):K(1278):K(1279):K
(1280):K(1281):K(1282):K(1283):K(1284):K
(1285):K(1286):K(1287):K(1288):K(1289):K
(1290):K(1291):K(1292):K(1293):K(1294):K
(1295):K(1296):K(1297):K(1298):K(1299):K
(1300):K(1301):K(1302):K(1303):K(1304):K
(1305):K(1306):K(1307):K(1308):K(1309):K
(1310):K(1311):K(1312):K(1313):K(1314):K
(1315):K(1316):K(1317):K(1318):K(1319):K
(1320):K(1321):K(1322):K(1323):K(1324):K
(1325):K(1326):K(1327):K(1328):K(1329):K
(1330):K(1331):K(1332):K(1333):K(1334):K
(1335):K(1336):K(1337):K(1338):K(1339):K
(1340):K(1341):K(1342):K(1343):K(1344):K
(1345):K(1346):K(1347):K(1348):K(1349):K
(1350):K(1351):K(1352):K(1353):K(1354):K
(1355):K(1356):K(1357):K(1358):K(1359):K
(1360):K(1361):K(1362):K(1363):K(1364):K
(1365):K(1366):K(1367):K(1368):K(1369):K
(1370):K(1371):K(1372):K(1373):K(1374):K
(1375):K(1376):K(1377):K(1378):K(1379):K
(1380):K(1381):K(1382):K(1383):K(1384):K
(1385):K(1386):K(1387):K(1388):K(1389):K
(1390):K(1391):K(1392):K(1393):K(1394):K
(1395):K(1396):K(1397):K(1398):K(1399):K
(1400):K(1401):K(1402):K(1403):K(1404):K
(1405):K(1406):K(1407):K(1408):K(1409):K
(1410):K(1411):K(1412):K(1413):K(1414):K
(1415):K(1416):K(1417):K(1418):K(1419):K
(1420):K(1421):K(1422):K(1423):K(1424):K
(1425):K(1426):K(1427):K(1428):K(1429):K
(1430):K(1431):K(1432):K(1433):K(1434):K
(1435):K(1436):K(1437):K(1438):K(1439):K
(1440):K(1441):K(1442):K(1443):K(1444):K
(1445):K(1446):K(1447):K(1448):K(1449):K
(1450):K(1451):K(1452):K(1453):K(1454):K
(1455):K(1456):K(1457):K(1458):K(1459):K
(1460):K(1461):K(1462):K(1463):K(1464):K
(1465):K(1466):K(1467):K(1468):K(1469):K
(1470):K(1471):K(1472):K(1473):K(1474):K
(1475):K(1476):K(1477):K(1478):K(1479):K
(1480):K(1481):K(1482):K(1483):K(1484):K
(1485):K(1486):K(1487):K(1488):K(1489):K
(1490):K(1491):K(1492):K(1493):K(1494):K
(1495):K(1496):K(1497):K(1498):K(1499):K
(1500):K(1501):K(1502):K(1503):K(1504):K
(1505):K(1506):K(1507):K(1508):K(1509):K
(1510):K(1511):K(1512):K(1513):K(1514):K
(1515):K(1516):K(1517):K(1518):K(1519):K
(1520):K(1521):K(1522):K(1523):K(1524):K
(1525):K(1526):K(1527):K(1528):K(1529):K
(1530):K(1531):K(1532):K(1533):K(1534):K
(1535):K(1536):K(1537):K(1538):K(1539):K
(1540):K(1541):K(1542):K(1543):K(1544):K
(1545):K(1546):K(1547):K(1548):K(1549):K
(1550):K(1551):K(1552):K(1553):K(1554):K
(1555):K(1556):K(1557):K(1558):K(1559):K
(1560):K(1561):K(1562):K(1563):K(1564):K
(1565):K(1566):K(1567):K(1568):K(1569):K
(1570):K(1571):K(1572):K(1573):K(1574):K
(1575):K(1576):K(1577):K(1578):K(1579):K
(1580):K(1581):K(1582):K(1583):K(1584):K
(1585):K(1586):K(1587):K(1588):K(1589):K
(1590):K(1591):K(1592):K(1593):K(1594):K
(1595):K(1596):K(1597):K(1598):K(1599):K
(1600):K(1601):K(1602):K(1603):K(1604):K
(1605):K(1606):K(1607):K(1608):K(1609):K
(1610):K(1611):K(1612):K(1613):K(1614):K
(1615):K(1616):K(1617):K(1618):K(1619):K
(1620):K(1621):K(1622):K(1623):K(1624):K
(1625):K(1626):K(1627):K(1628):K(1629):K
(1630):K(1631):K(1632):K(1633):K(1634):K
(1635):K(1636):K(1637):K(1638):K(1639):K
(1640):K(1641):K(1642):K(1643):K(1644):K
(1645):K(1646):K(1647):K(1648):K(1649):K
(1650):K(1651):K(1652):K(1653):K(1654):K
(1655):K(1656):K(1657):K(1658):K(1659):K
(1660):K(1661):K(1662):K(1663):K(1664):K
(1665):K(1666):K(1667):K(1668):K(1669):K
(1670):K(1671):K(1672):K(1673):K(1674):K
(1675):K(1676):K(1677):K(1678):K(1679):K
(1680):K(1681):K(1682):K(1683):K(1684):K
(1685):K(1686):K(1687):K(1688):K(1689):K
(1690):K(1691):K(1692):K(1693):K(1694):K
(1695):K(1696):K(1697):K(1698):K(1699):K
(
```


ATAROWCY

o sobie

W drugim wydaniu specjalnym „Bajtki” — „Tylko o Atari” opublikowana została ankieta dla czytelników. Pierwsze odpowiedzi nadeszły już po dwóch dniach od ukazania się pisma w kioskach. Łącznie otrzymaliśmy do końca lutego 1989 roku ponad 3000 listów — dokładnie 3004 — i tyle właśnie zostało uwzględnionych przy opracowywaniu wyników ankiety.

Przeciętny czytelnik „Bajtki-Atari” jest mężczyzną w wieku 14—15 lat, uczącym się w szkole podstawowej. Mieszka w dużym mieście (ponad pół miliona mieszkańców) — najczęściej na Śląsku albo w województwie warszawskim, gdańskim lub krakowskim. Posiada on komputer Atari 65XE z magnetofonem firmowym, a jako monitora używa czarno-białego telewizora lub monitora monochromatycznego. Przyjrzyjmy się bliżej, co składa się na taki obraz czytelnika.

Czytelnicy

Naszymi czytelnikami są w przytłaczającej większości mężczyźni, kobiety stanowią bowiem tylko 3,8% respondentów. Bardzo szeroki jest natomiast zakres wieku. Najmłodsi czytelnicy mają 8 lat, a najstarszy 72 lata. Przeważa jednak młodzież — 78,5% nie ukończyło jeszcze 18 lat. Szczegółowo rozkład wieku pokazuje wykres na rys. 1.

Ponad połowa odpowiedzi (52,4%) pochodziła od uczniów szkół podstawowych, a prawie jedna czwarta (23,6%) od uczniów szkół średnich. Spośród pozostałych grup najliczniejsze są osoby posiadające średnie wykształcenie techniczne (6,1%). Trochę dziwi nas niewielka liczba studentów — tylko 3,7%. Prawdopodobnie zajmują się oni komputerami uważanymi powszechnie (ale niekoniecznie słusznie) za bardziej „profesjonalne”. Wykształcenie czytelników jest przedstawione na rys. 2.

Jak łatwo można było przewidzieć, większość czytelników (23,8%) mieszka w dużych miastach, liczących ponad pół miliona mieszkańców. Pew-

nym zaskoczeniem jest jednak dosyć równomierny rozkład w pozostałych miastach — od 12,0 do 14,3% (rys. 3). Wyraźnie mniejsze jest tylko zainteresowanie naszym pismem na wsi i w małych miasteczkach. Prawdopodobnie wynika to z utrudnionego dostępu do „Bajtki”.

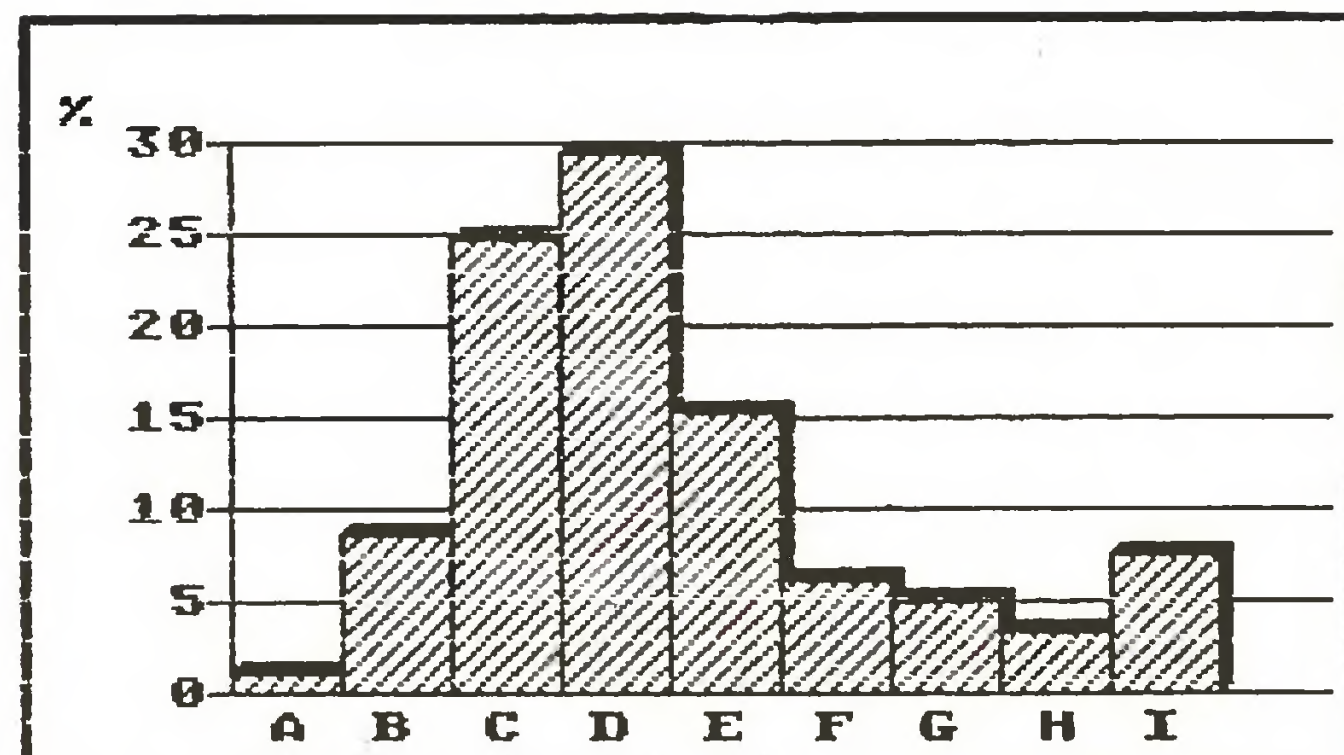
Odpowiedzi otrzymaliśmy z całej Polski, a także z zagranicy (głównie CSRS). Najdłuższą drogę przebył list od Czytelnika z Danii, a najkrótszą — od sąsiada mieszkającego nad Wojtkiem. Najwięcej ankiet, bo aż 13,0%, nadeszło z województwa katowickiego, co jest zrozumiałe, ze względu na największą liczbę ludności. Dalsze miejsca zajęły województwa: stołeczne (9,8%), gdańskie (6,8%) oraz krakowskie (6,2%). Najmniej czytelników „Bajtki-Atari” mieszka w województwach: chełmskim (0,2%) oraz zamojskim, sieradzkim i leszczyńskim (po 0,3%). Na rysunku 4 pokazane są liczby ankiet, które zostały wysłane z poszczególnych województw.

Według podanych przez czytelników informacji o posiadanych komputerach przeprowadziliśmy ciekawe obliczenia. Biorąc pod uwagę liczbę komputerów sprzedanych przez Pewex ustaliliśmy, ile zostało przywiezionych prywatnie. Następnie określiliśmy przybliżoną liczbę komputerów znajdujących się w poszczególnych województwach. Po wykonaniu prostego dzielenia otrzymaliśmy rezultat przedstawiony na rysunku 5 — liczbę Atari przypadającą na 1000 mieszkańców. Warto dobrze się przyjrzeć temu rysunkowi, gdyż widać tam bardzo dziwne rzeczy. Duża ilość komputerów na Śląsku nikogo nie zaskakuje, ale w Suwałkach i Białymstoku! Tak zwana „Polska B” przesunęła się chyba trochę na południe i na zachód.

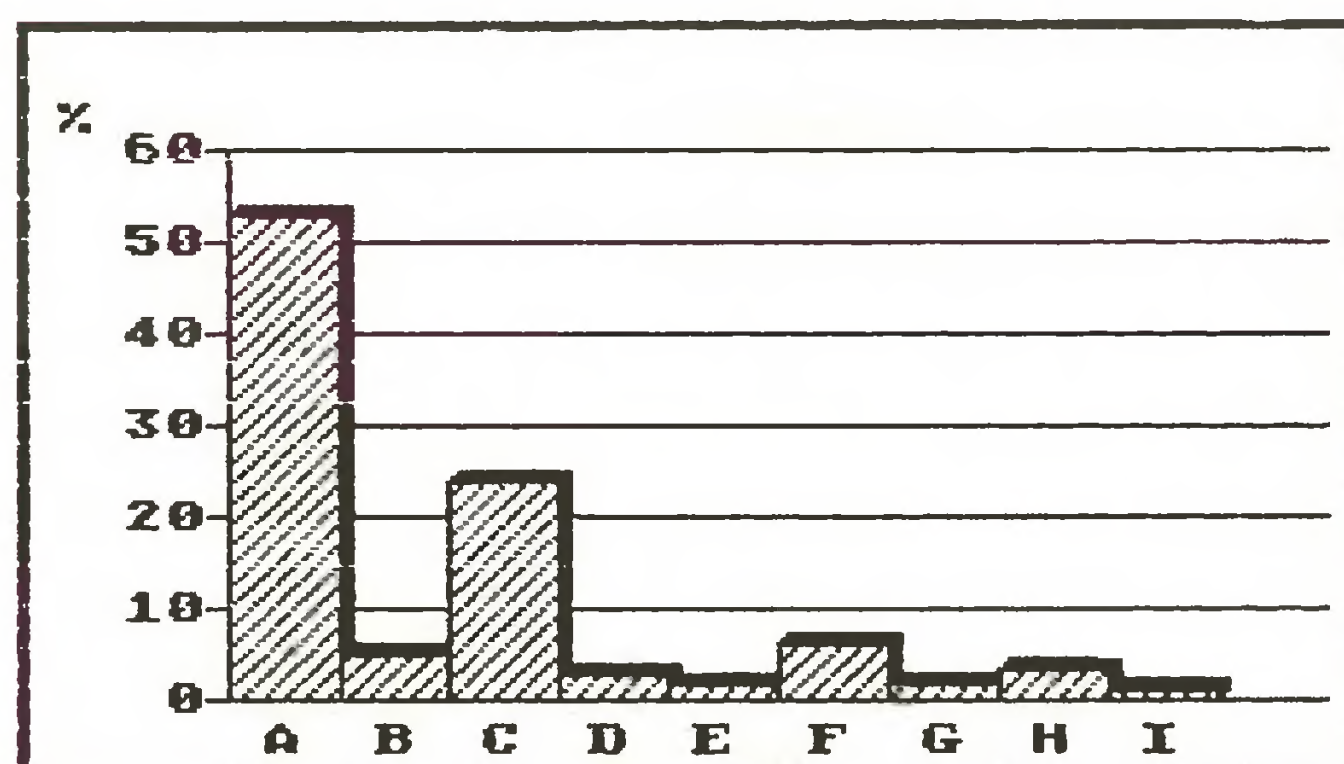
Sprzęt

Własne komputery (oczywiście Atari) posiada 95,9% respondentów. W przytłaczającej większości są to komputery ośmiobitowe — tylko 1,5% czytelników jest właścicielami Atari ST. Spośród modeli XL i XE 87,5% ma 64 KB pamięci (Atari 65XE i 800XE — 60,4%, Atari 800XL — 27,1%), a większą pamięcią dysponuje 12,3%. Najmniejszy komputer (Atari 600XL) występuje w ilości śladowej. Typy komputerów są pokazane na rys. 6. Prawie trzy czwarte (71,7%) Atari zostało zakupione w Pewexie, a 14,9% za granicą. Pozostali użytkownicy nabyli swój sprzęt w Polsce, głównie na różnych giełdach.

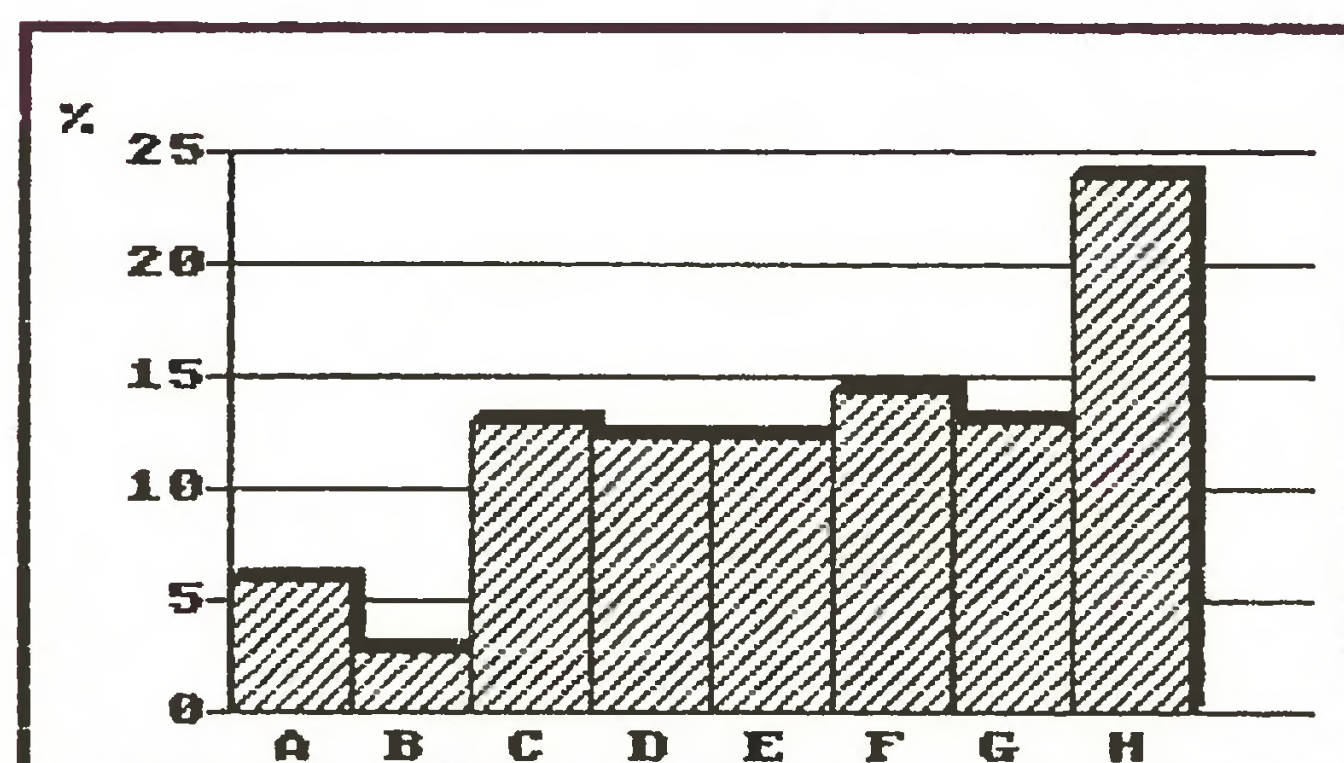
Bardzo małe jest zróżnicowanie wykorzystywanych pamięci masowych. Aż 83,3% czytelników używa tylko magnetofonów, w tym 10,0% magnetofonów nie przeznaczonych do komputerów. Sta-



Rys. 1. Wiek czytelników A — poniżej 10 lat, B — 10—11 lat, C — 12—13 lat, D — 14—15 lat, E — 16—17 lat, F — 18—19 lat, G — 20—24 lata, H — 25—29 lat, I — ponad 30 lat



Rys. 2. Wykształcenie czytelników A — uczniowie szk. podst., B — uczniowie szk. zawod., C — uczniowie szk. śred., D — studenci kier. techn., E — studenci kier. human., F — wyksz. średnie techn., G — wyksz. wyższe techn., I — wyksz. wyższe human.

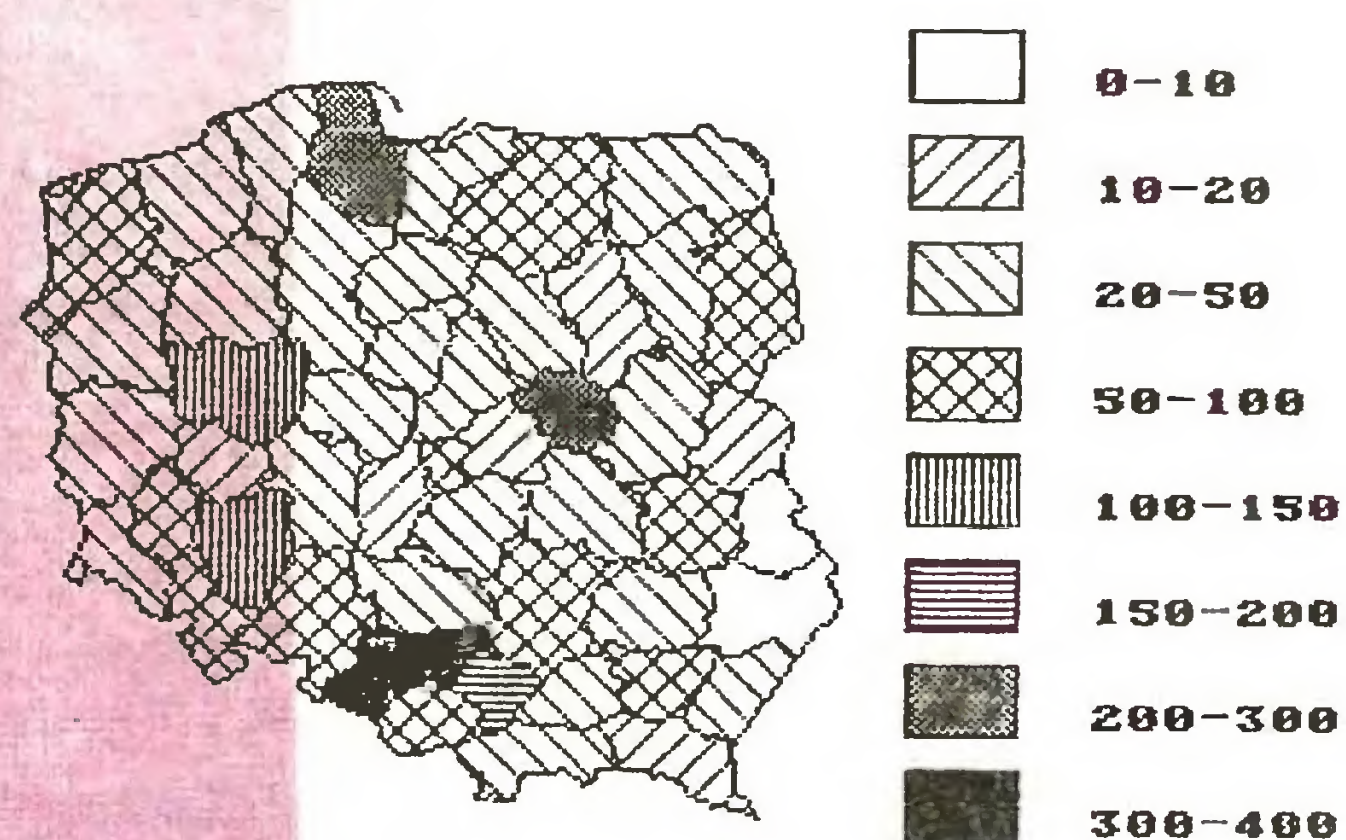


Rys. 3. Miejsce zamieszkania czytelników A — wieś, B — miasto do 5 tys. mieszk., C — miasto 5—25 tys. mieszk., D — miasto 25—50 tys. mieszk., E — miasto 50—100 tys. mieszk., F — miasto 100—250 tys. mieszk., G — miasto 250—500 tys. mieszk., H — miasto ponad 500 tys. mieszk.

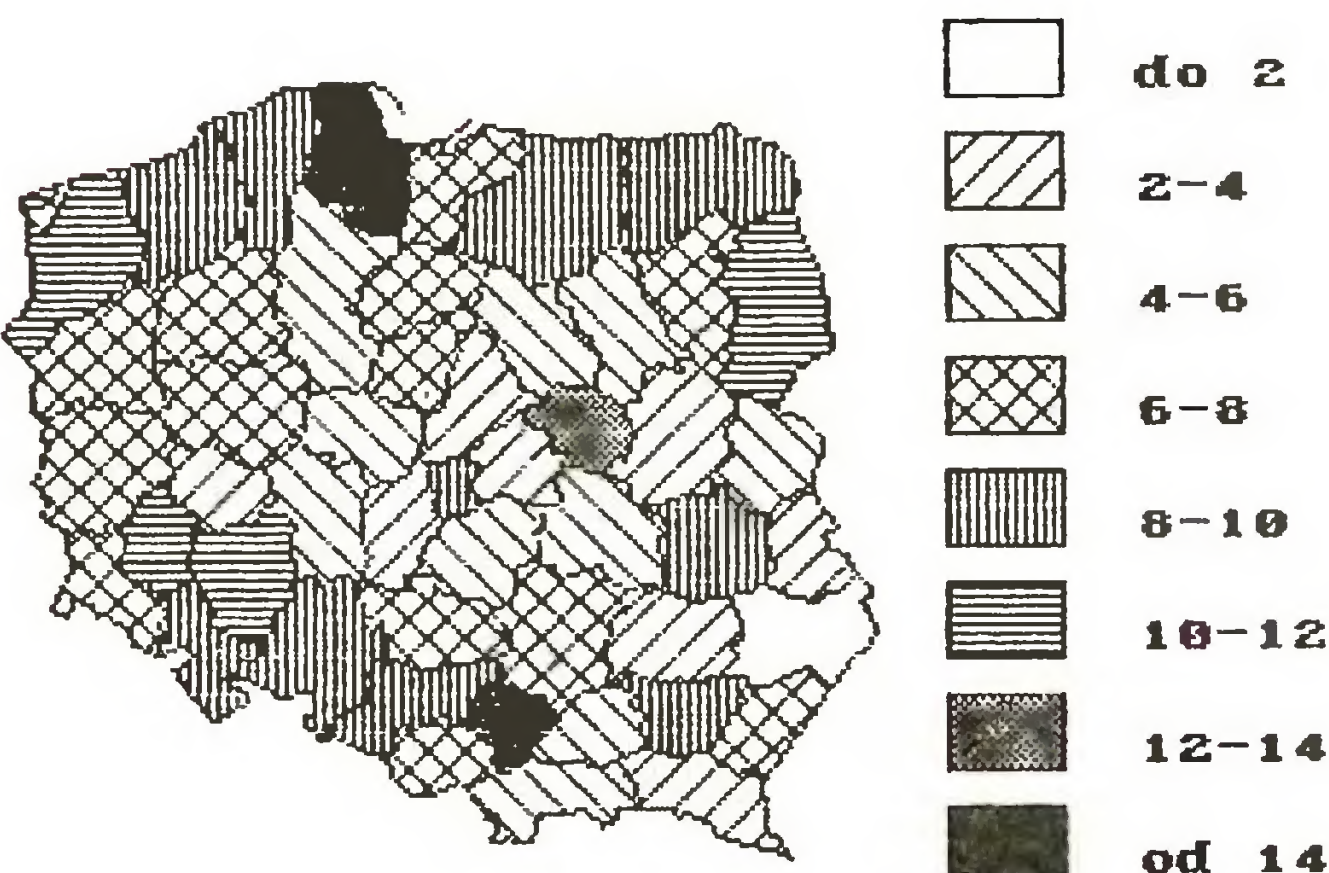
cię dysków posiada 16,4% respondentów, a ponad połowa z nich (55,9%) ma również magnetofon, choć przypuszczamy, że wykorzystują go sporadycznie. Razem więc magnetofony są w posiadaniu 92,7% czytelników. Rodzaje pamięci masowych są przedstawione na rys. 7.

Kolejne pytanie ankiety dotyczyło rodzaju monitora. Z odpowiedzi na nie (rys. 8) widać, że mało kolorowy obraz — telewizory czarno-białe wykorzystuje 37,0%, telewizory w systemie SECAM 8,4%, a monitory monochromatyczne 30,4% czytelników. Tylko jedna czwarta użytkowników może cieszyć się kolorami na ekranie (telewizory PAL 20,5%, monitory kolorowe 3,7%). Nasuwa się pytanie, po co aż 256 kolorów w komputerze, skoro tak niewiele osób je ogląda.

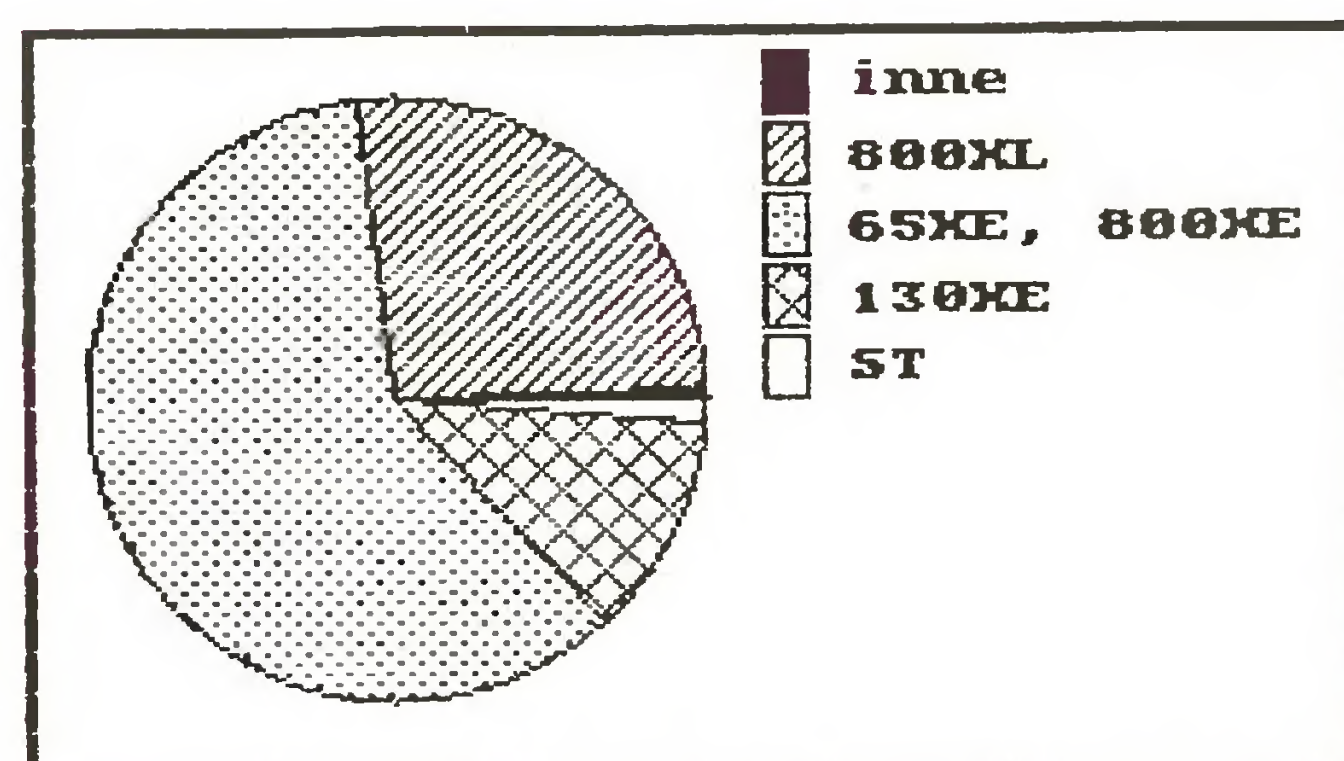
Trzecim urządzeniem peryferyjnym decydującym o poważnym zastosowaniu komputera jest



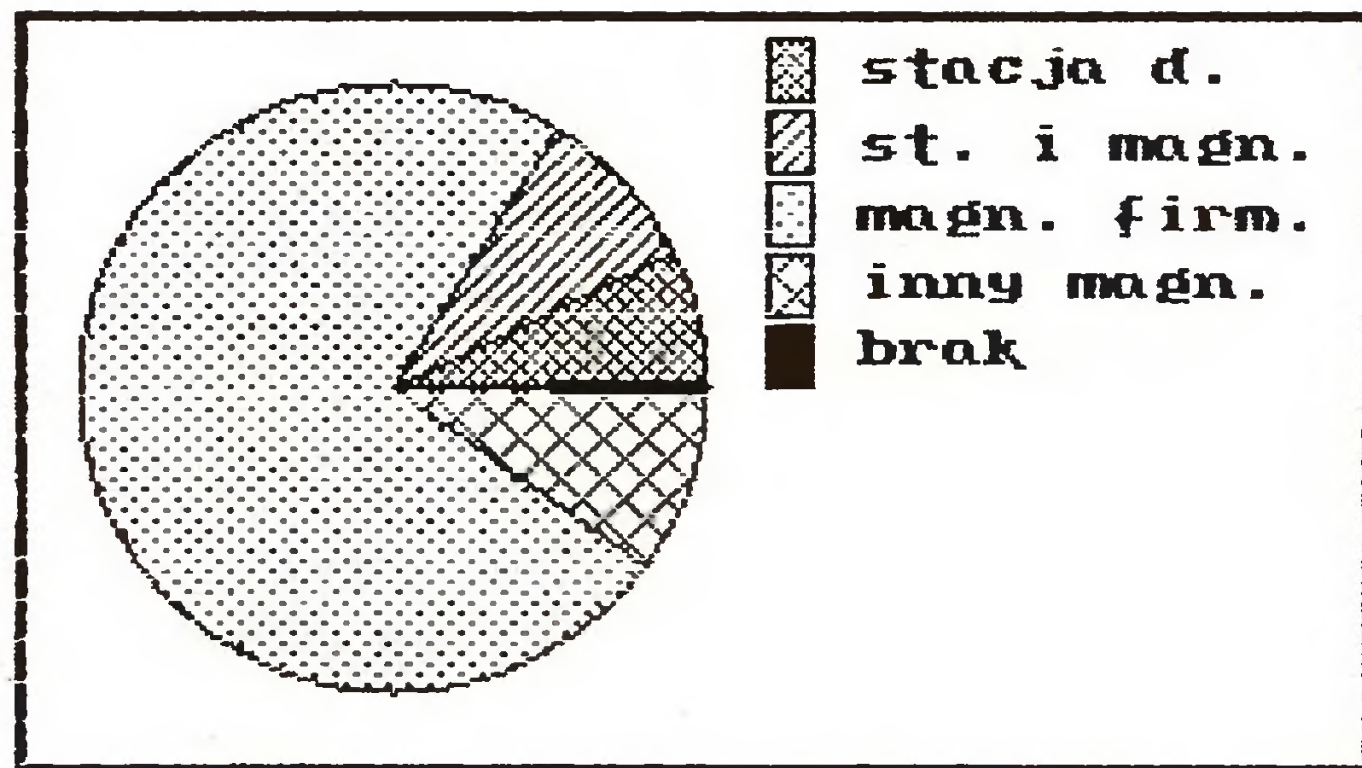
Rys. 4. Liczba ankiet nadesłanych z poszczególnych województw



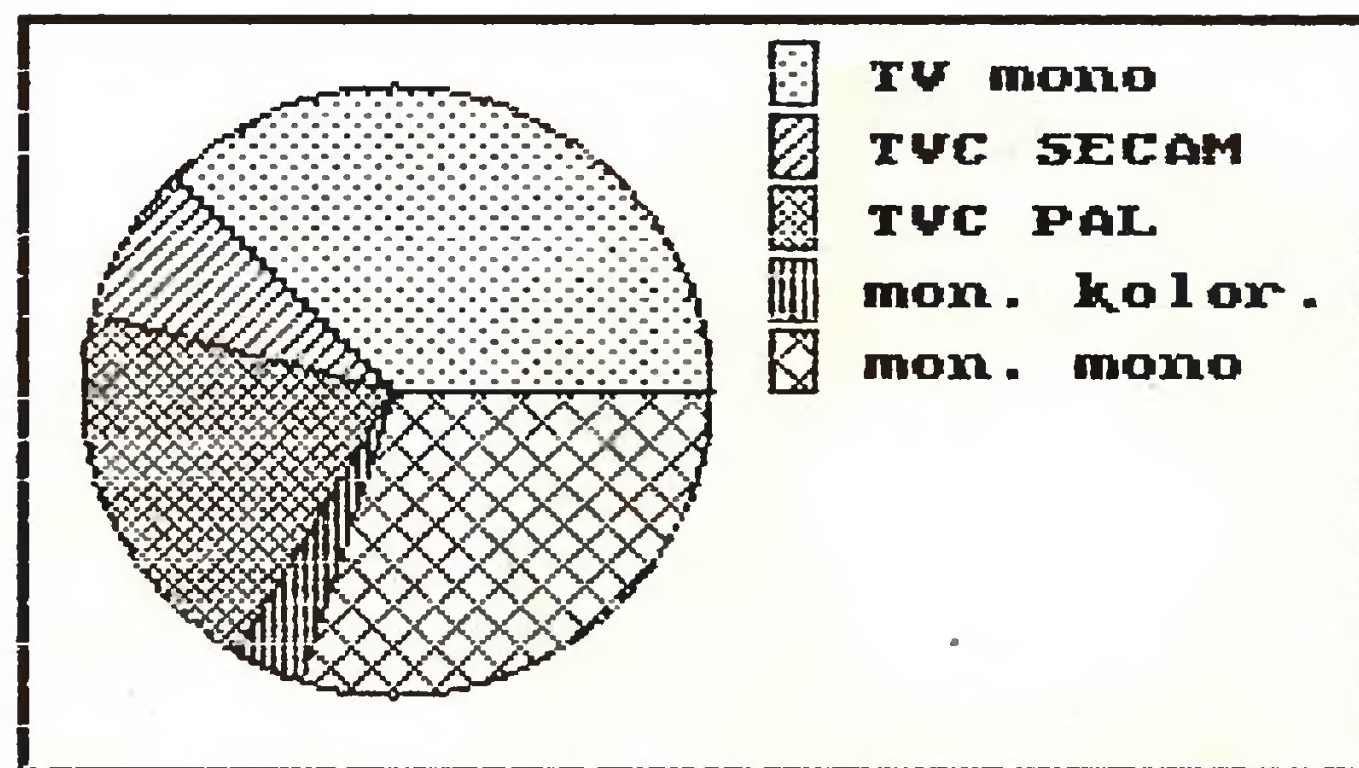
Rys. 5. Liczba komputerów Atari na 1000 mieszkańców wg województw



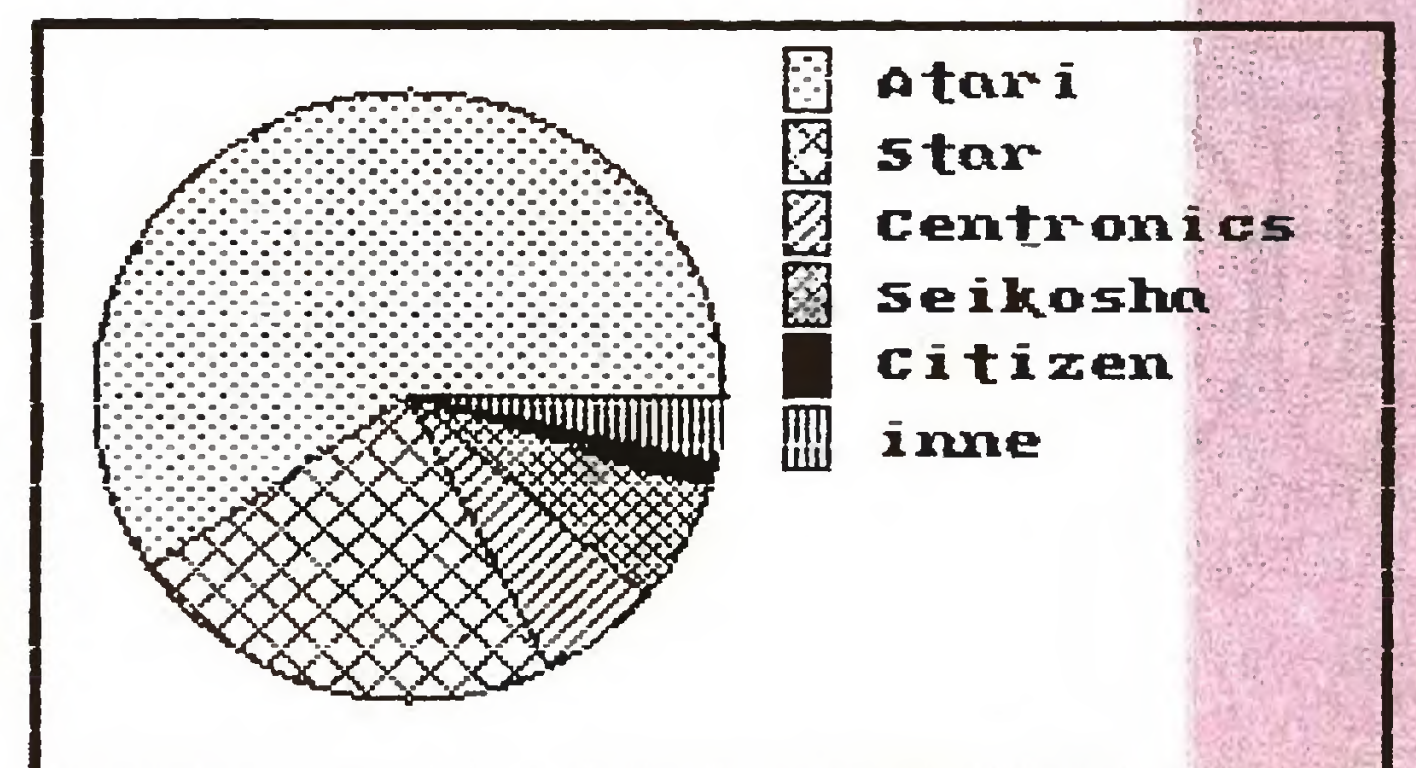
Rys. 6. Komputery Atari posiadane przez czytelników



Rys. 7. Pamięci masowe posiadane przez czytelników



Rys. 8. Monitory posiadane przez czytelników



Rys. 9. Typy drukarek posiadane przez czytelników

drukarka. Można powiedzieć, że tu sytuacja jest tragiczna, posiada ją bowiem tylko 4,7% użytkowników. Na domiar złego dominują wśród nich przestarzałe drukarki Atari (58,8%), które były sprzedawane w Pewexie. Drugie miejsce zajmuje renomowany Star (22,8%). Ze względu na konieczność dokonania zakupu za granicą drukarki Citizen stanowią tylko 1,5%, pomimo iż za cenę Atari dają możliwości Stara. Rodzaje drukarek współpracujących z Atari są pokazane na rys. 9.

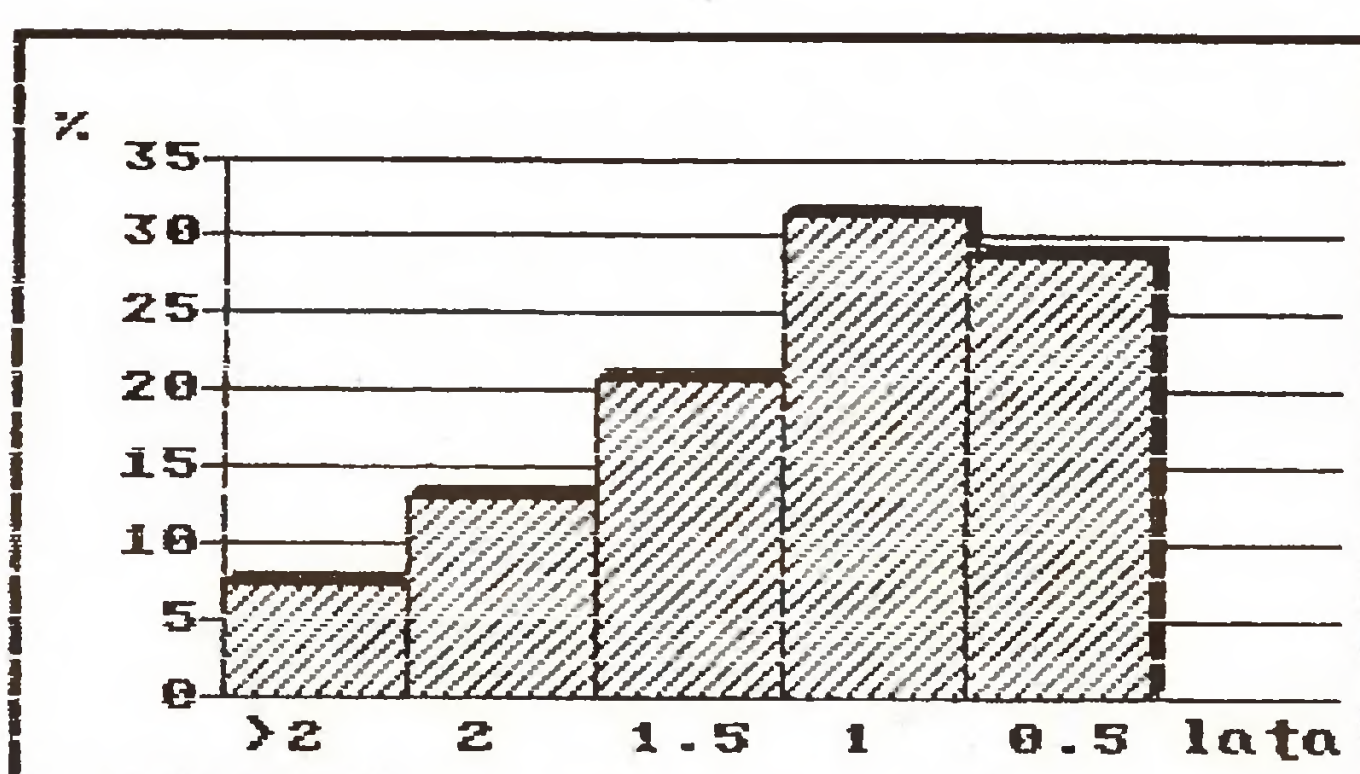
Komputer jest stosunkowo nowym nabytkiem w każdym domu. Tylko 7,2% czytelników posiada go dłużej niż dwa lata. Większość (59,6%) cieszy się nim dopiero od roku lub jeszcze krócej. Wykres przedstawiony na rys. 10 pokazuje, jak szybko wzrastały zakupy sprzętu. Nikt nie mógł tego przewidzieć i nawet w Pewexie komputerów... zabrakło. To załamanie widać wyraźnie na wykresie. Niewielka grupa odpowiedzi (7,2%) pochodzi od osób, które miały przedtem inny komputer. Bardzo interesujący jest przegląd typów tych komputerów. Króluje wśród nich Spectrum, które stanowi 53,6%. Dalsze miejsca zajmują kolejno: ZX-81 (15,0%), C-16 (9,7%) i różne modele Amstrada (4,8%).

To jednak już historia, a co dalej? Dla 37,6% czytelników nic, gdyż nie planują oni dalszych zakupów. W planowanych nabytkach dominuje zaś stacja dysków (71,0%). Potem długo, długo nic i na drugim miejscu są wymieniane drukarki (26,0%). Z pozostałych urządzeń największym popytem będą się cieszyć monitory (6,8%), a inne peryferia występują w bardzo małych ilościach (poniżej 4%). Biorąc pod uwagę wiek odpowiadających trzeba szczerze powiedzieć, że są to raczej chęci niż rzeczywiste plany, gdyż sprzęt komputerowy jest w Polsce drogi.

Zastosowania

Opisana wyżej struktura sprzętu nasuwa smutny wniosek, że większość komputerów jest traktowana jako zabawki. Zestaw złożony z komputera i magnetofonu nie nadaje się do prawie żadnej pracy i trudno mówić w takim przypadku o poważniejszych zastosowaniach. Do tego potrzebna jest co najmniej stacja dysków, a i tak brak drukarki poważnie utrudnia pracę. W krajach rozwiniętych podstawowy zestaw musi zawierać (oprócz komputera) monitor, stację dysków, drukarkę, a najczęściej także modem.

Zabawowe wykorzystanie komputerów znalazło swoje odbicie w odpowiedzi na pytanie o najczęściej używane programy (rys. 11). Dla większości czytelników są to gry zręcznościowe (58,1%), przygodowe (57,7%) i symulacyjne (40,3%). Natomiast gry strategiczne wymieniło tylko 23,8% użyt-



Rys. 10. Czas posiadania komputera

kowników. Widocznie zbyt dużo trzeba przy nich myśleć. Także wskazywane programy użytkowe świadczą o rozrywkowym traktowaniu Atari. Najwięcej głosów otrzymały w tej grupie programy graficzne (37,9%) i muzyczne (36,9%). Programy służące do poważniejszych zastosowań uzyskały stosunkowo mało głosów: edytory tekstu — 4,4%, bazy danych — 6,8%, a programy obliczeniowe — 11,4%.

W przeciwieństwie do powyższych wypowiedzi, 56,0% ankietowanych podało, że programują czasami, a 18,0%, że często (języki programowania wymienili jednak tylko 14,8%). Obawiamy się, że niektórzy za programowanie uważają również przepisywanie programów z „Bajtki” i innych czasopism komputerowych. Są zapewne i tacy, którzy piszą programy złożone z dwóch-trzech wierszy i także uważają się za programistów. Świadczy o tym, między innymi, wymieniane języki programowania. Bezkonkurencyjny jest najgorszy, czyli Atari Basic, który wymieniło 85,7% osób programujących. Inne języki są używane znacznie rzadziej, co przedstawia rys. 12. Zaskoczyła nas tu wysoka pozycja Logo (17,7% odpowiedzi). Ostatnio poświęcamy mu niewiele miejsca na łamach „Bajtki” i trzeba to koniecznie zmienić. Dalsze, prawie równorzędne miejsca zajmują: Action! (8,1%), język maszynowy (7,5%), Turbo Basic (6,4%) i Pascal (6,3%).

„Bajtek”

Jednym z najważniejszych w ankiecie było pytanie o zawartość naszego pisma. Niektóre odpowiedzi wprawiły nas w zdumienie. Na przykład: za mało gier, a wszystkich pozostałych materiałów za dużo. Albo: dziewięcioletni chłopiec chciałby mniej gier, za to więcej kursów programowania i recenzji, a czterdziestoletni inżynier — odwrotnie.

Prawie wszyscy czytelnicy wskazywali, czego powinno być więcej. Niewiele osób pamiętało, że „Bajtek” nie jest z gumy. Wyraźnie widoczne jest to na rys. 13, gdzie kolejne części słupków (od dołu) pokazują, czego jest zbyt mało, czego akurat i czego zbyt dużo. Dla pełnego zaspokojenia życzeń czytelników pismo powinno mieć 45—50 stron. Niestety, jest to niemożliwe (choć też o tym marzymy).

Przejdźmy teraz do szczegółów. Najwięcej przeciwników mają reportaże i wywiady (37,0%) oraz opisy klubów (15,0%). W pozostałych grupach odpowiedzi „za dużo” było niewiele. Co zatem jest najchętniej czytane? Przede wszystkim programy i programowanie: programy użytkowe (58,0% odpowiedzi), gry (52,1%) i kursy programowania (51,1%). W następnej kolejności czytelnicy oczekują instrukcji (48,8%) i opisów (47,1%), programów użytkowych oraz opisów gier (43,3%). Dokładne wyniki są uwidocznione na rys. 13, a jego analiza powie więcej niż najdokładniejszy nawet opis.

Wyniki ankiety w znacznym stopniu potwierdziły przypuszczenia redakcji na temat czytelników „Bajtki”, ich sprzętu i zainteresowań. Niektóre nasze poglądy trzeba było jednak zweryfikować. Mamy nadzieję, że dzięki Waszej pomocy i współpracy „Bajtek” będzie jeszcze bardziej interesujący i chętniej czytany.

Ewa Mirosz-Molska
Marek Zachar

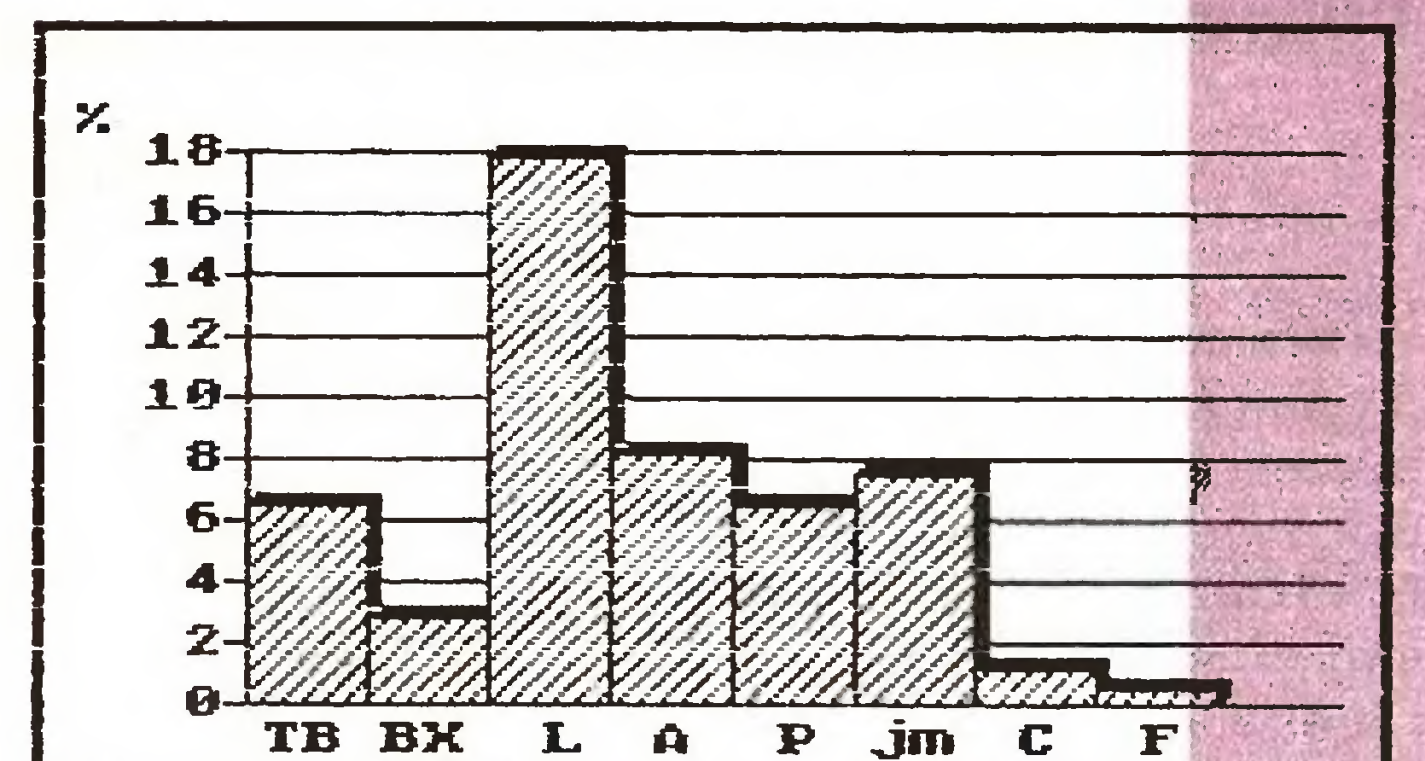
Kasety z programami ufundowane przez firmę „Intersoft”, 00—443 Warszawa, ul. Górnośląska 9/11, tel. 21-56-08, wylosowali:

1. Dariusz Domżała, Warszawa
2. Małgorzata Gawęda, Czaniewo
3. Maciej Grycz, Dobrzyńsk
4. Paweł Januszewski, Zgorzelec
5. Ryszard Jasiński, Czarny Dunajec
6. Krzysztof Rosa, Gdynia
7. Dawid Rzechaczek, Grodzisk Wlkp.
8. Grzegorz Sarnecki, Kamienna Góra
9. Jacek Szymanowski, Suwałki
10. Marcin Zdanowicz, Ostrołęka

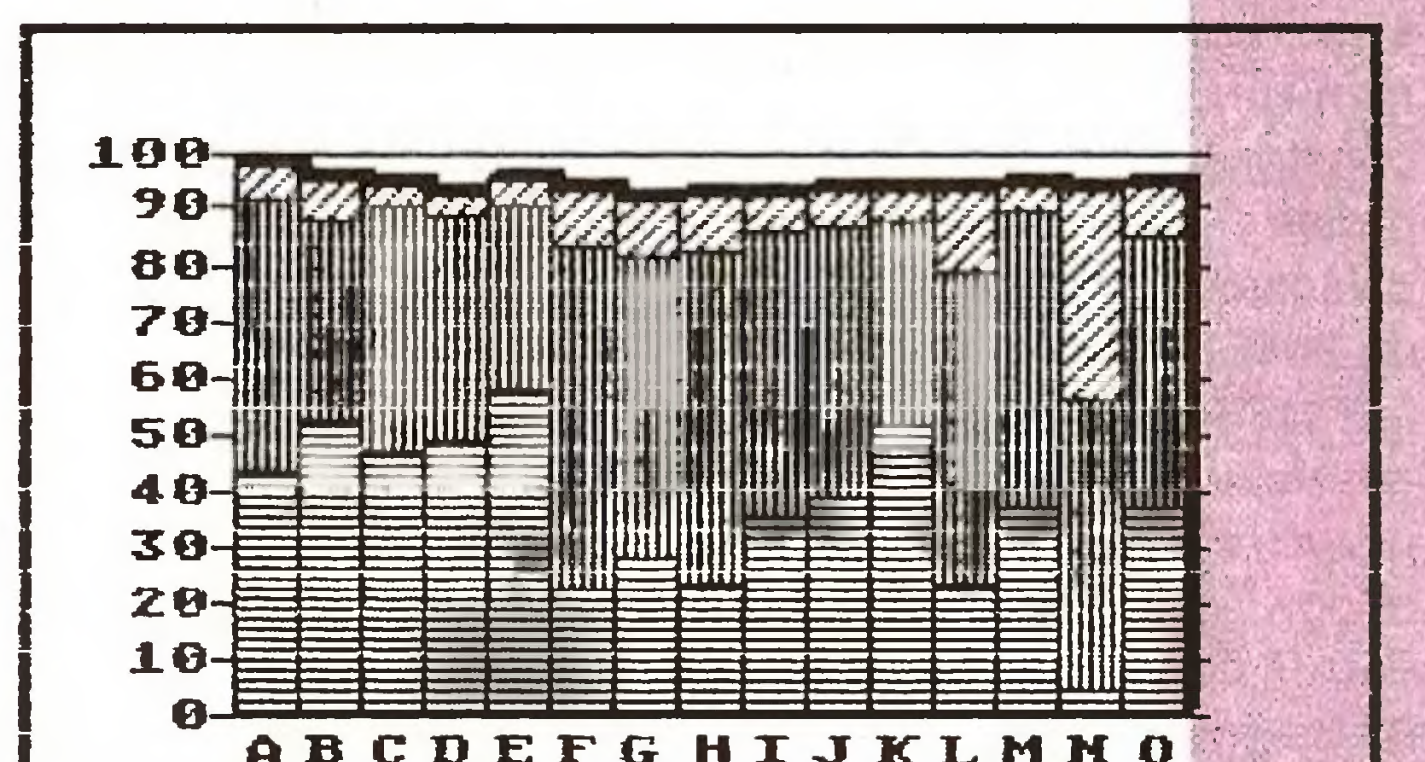
Prosimy wymienione wyżej osoby o skontaktowanie się z redakcją.



Rys. 11. Najczęściej używane programy A — gry zręcznościowe, B — gry strategiczne, C — gry przygodowe, D — gry symulacyjne, E — edytory tekstu, F — programy graficzne, G — programy muzyczne, H — programy obliczeniowe, I — programy edukacyjne, J — bazy danych, K — języki programowania, L — inne programy użytkowe.

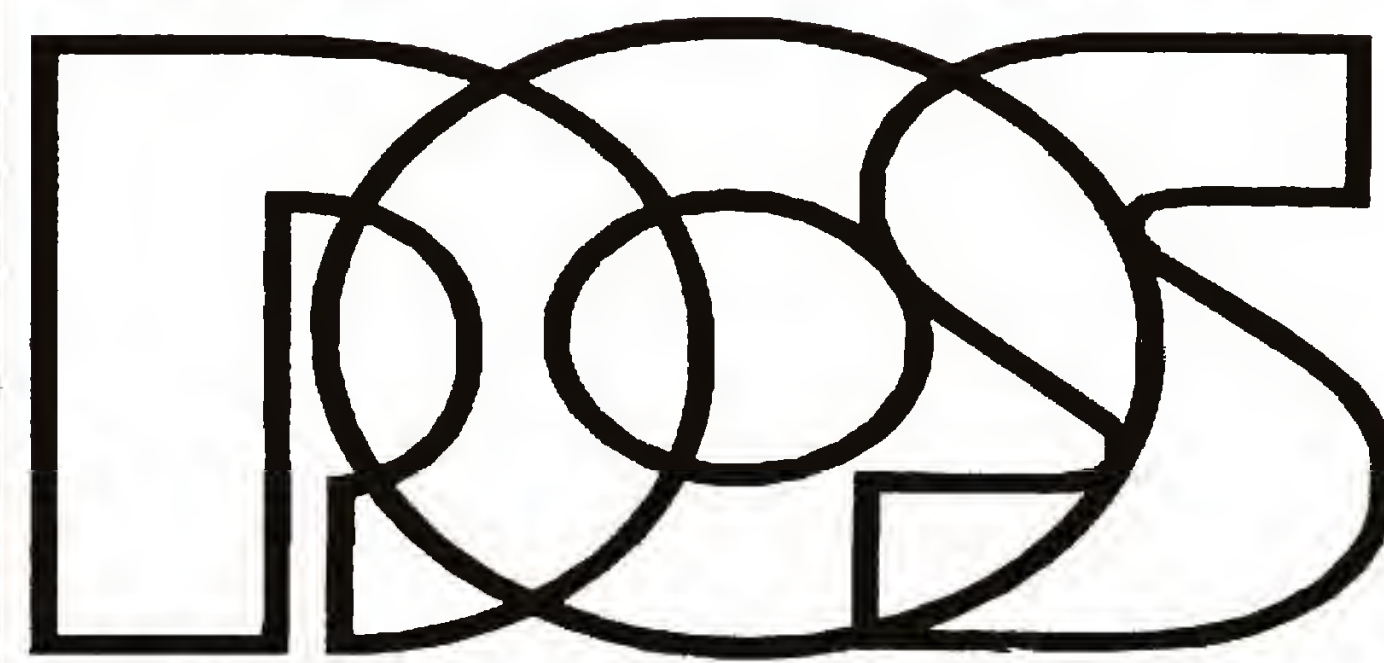


Rys. 12. Najczęściej stosowane języki programowania (poza Atari Basic) TB — Turbo Basic, BX — Basic XL/XE, L — Logo, A — Action!, P — Pascal, jm — język maszynowy, C — C, F — Forth.



Rys. 13. Czego jest: za mało, akurat i za dużo A — opisy gier, B — wydruki gier, C — opisy programów użytkowych, D — instrukcje programów użytkowych, E — wydruki pogromów użytkowych, F — opisy sprzętu, G — modyfikacje sprzętu, H — recenzje książek, I — recenzje programów, J — opisy języków programowania, K — kursy programowania, L — opisy i adresy klubów, M — nowości rynkowe, N — wywiady i reportaże, O — podstawy działania komputerów i peryferii.

MULTI



Niektóre programy w wersji kasetowej działają ze stacji dysków wyłącznie za pomocą programu „MultiBoot”.

Wykorzystywanie ich poprzez DOS jest niemożliwe. Przekształcenie takiego programu na formę czytelną dla DOS-u jest bardzo pracochłonne. Zamieszczony tu program „MultiBoot to DOS” (w skrócie „MultiDOS”) służy do automatycznego wykonywania takiej konwersji.

UWAGA: program ten jest przeznaczony wyłącznie dla posiadaczy stacji dysków.

Pierwotnie program napisano w assemblerze MAC/65. Aby ułatwić wpisywanie, został następnie przetłumaczony na ciąg liczb i w takiej postaci jest tu opublikowany.

W celu otrzymania programu „MultiDOS” należy przepisać zamieszczony wydruk programu w Basicu i uruchomić go. Jeśli podczas wpisywania popełniliśmy błąd, komputer wyświetli numer wiersza, w którym jest pomyłka, i program zostanie zatrzymany. Należy wtedy poprawić błąd i ponownie uruchomić program. Gdy nie będzie żadnych błędów, komputer wyświetli odpowiedni komunikat. Należy wtedy umieścić w stacji numer 1 dyskietkę sformatowaną uprzednio za pomocą DOS-u i nacisnąć (RETURN). Właściwy program „MultiDOS” zostanie zapisany na dyskietce i można już z niego korzystać wczytując za pomocą funkcji „L” DOS-u.

Program „MultiDOS” zawiera pełne komunikaty o błędach i objaśnienia dla użytkownika. Na początku wybieramy sposób odczytu przetwarzanego programu:

- D — odczyt pliku z dyskietki „MultiBoot”
- B — odczyt programu całodyskowego (boot disk)
- C — odczyt jednoblokowego programu z kasy.

Przez program całodyskowy należy przy tym rozumieć program jednoblokowy o identycznej budowie jak jednoblokowy program kasetowy, lecz zapisany na dyskietce od jej pierwszego sektora i po 128 (lub 256) bajtów w sektorze. Po wybraniu właściwego wariantu odczytujemy program, postępując według wyświetlanych komunikatów. Po zakończeniu odczytu ukaże się menu bardzo podobne do menu „FCopy”. Jedyną różnicą jest napis OPERATION STATUS, po którym znajdują się litery OK. Oznacza to, że wszystko jest w porządku. Gdy zdarzy się jakiś błąd, to jego numer zostanie umieszczony na miejscu napisu OK.

Menu zawiera następujące warianty:

- # — skasowanie butora i ponowne uruchomienie „MultiDOS”
- *n — katalog dyskietki w stacji dysków numer n
- @ — uruchomienie odczytanego programu
- Fn — formatowanie dyskietki w stacji dysków numer n
- D — powrót do DOS-u (koniec programu)

Należy wybrać jeden z wymienionych wariantów lub podać specyfikację pliku, w którym będzie zapisany przekształcony program, np. D:NOWAGRA.COM. Tak zapisany program można uruchomić zarówno za pomocą DOS-u, jak i dowolnego inicjalizera.

UWAGA: program „MultiDOS” nie jest odporny na RESET.

Nie publikujemy wydruku programu źródłowego, gdyż zajmuje on ponad 20 stron formatu A4. Jeśli ktoś chciałby zbadać program, to trzeba wykonać disasemblację albo napisać do autora za pośrednictwem redakcji. Dla ewentualnych „szperaczy” przydatna będzie informacja, że program można podzielić na trzy części:

- wiersze 100—121: czołówka programu
- wiersze 122—358: właściwy program
- wiersze 359—428: komunikaty

Cały program zajmuje 3288 bajtów, czyli 27 sektorów na dyskietce. Przypominam, że musi być wczytywany przy pomocy DOS-u. Programu tego używam od dziesięciu miesięcy i, jak dotąd, nie spotkałem programu, który po konwersji nie uruchamiałby się, a miałem ich ponad czterdzieści.

```

0100 PMG = 20480 ;początek
0110 PX = 48 ;obszaru PMG
0120 PY = 16 ;(wielokrotnosc
0130 POM = 203 ;2K)
0140 POM1 = 205
0150 PMBASE = 54279
0160 DMACNTL = 559
0170 PMCNTL = 53277
0180 POSTACI = PMG+256
0190 PLYR1 = PMG+512
0200 PLYR3 = PLYR1+256
0210 PLYR4 = PLYR3+128
0220 HPOS = 53248
0230 SETVBLNK = 58460
0240 EXITVBLNK = 58466
0250 Y = X+4
0260 G = X+8
0270 ;
0280 *= PMG
0290 X .BYTE 0,0,0,0,0,0
0300 .BYTE 0,0,0,0,0,0
0310 ;
0320 PLA ;ustawienie
0330 LDA #7 ;VBLANK
0340 LDY # <PRZERWANIE
0350 LDX # >PRZERWANIE
0360 JMP SETVBLNK
0370 ;
0380 PRZERWANIE
0390 LDA #46 ;wpisanie
0400 STA DMACNTL ;wartosci
0410 LDA # >PMG ;wymaganych
0420 STA PMBASE ;do pojawienia
0430 LDA #3 ;sie grafiki
0440 STA PMCNTL ;graczy
0450 LDA #0
0460 TAX
0470 HOP1
0480 STA PLYR1,X
0490 INX ;zerowanie
0500 BNE HOP1 ;obszaru
0510 HOP2
0520 STA PLYR3,X ;zajmowanego
0530 INX ;przez obrazy
0540 BNE HOP2 ;graczy
0550 ;
0560 STA POM1
0570 LDA # <PLYR1
0580 STA POM
0590 LDA # >PLYR1
0600 STA POM+1
0610 ;
0620 PETLA
0630 LDX POM1
0640 LDA X,X ;sprawdzenie
0650 BEQ KNPTL ;czy gracz
0660 CLC ;bierze
0670 ADC #PX ;udzial w grze
0680 STA HPOS,X
0690 LDA Y,X
0700 CLC
0710 ADC POM
0720 ADC #PY
0730 STA POM
0740 LDA G,X ;jesli tak, to
0750 ASL A ;przepisz
0760 ASL A ;odpowiednia
0770 ASL A ;postac
0780 ASL A
0790 TAX
0800 JSR PRZEPISUJ
0810 KNPTL
0820 LDA POM
0830 AND #128
0840 EOR #128
0850 STA POM
0860 BMI SPRAWDZ
0870 INC POM+1
0880 SPRAWDZ
0890 INC POM1
0900 LDA POM1
0910 CMP #4
0920 BNE PETLA
0930 JSR WSTAWKA
0940 JMP EXITVBLNK
0950 ;
0960 PRZEPISUJ
0970 LDY #15
0980 PRP
0990 LDA POSTACI,X ; procedura
1000 STA (POM),Y ; przenoszaca
1010 INX ; wybrana
1020 DEY ; postac
1030 BPL PRP ; dla danego
1040 RTS ; gracza
1050 ;
1060 WSTAWKA
1070 RTS

```

Listing 3

```

1060 JOY1 = 632
1070 WSTAWKA
1080 LDA JOY1
1090 LSR A
1100 BCS BIT1
1110 DEC Y
1120 BIT1
1130 LSR A
1140 BCS BIT2
1150 INC Y
1160 BIT2
1170 LSR A
1180 BCS BIT3
1190 DEC X
1200 BIT3
1210 LSR A
1220 BCS KONSPR
1230 INC X
1240 KONSPR
1250 LDA X
1260 BNE XWI
1270 INC X
1280 XWI
1290 CMP #160
1300 BNE XMN
1310 DEC X
1320 XMN
1330 LDA Y
1340 BNE YWI
1350 INC Y
1360 YWI
1370 CMP #96
1380 BNE KONWST
1390 DEC Y
1400 KONWST
1410 RTS

```

Listing 4

postaci odpowiadającej danemu duszkowi. Można tam wpisywać wartości od 0 do 15. Dzięki trzem rejestrom dla każdego duszka można uzyskać zarówno płynny, jak i skokowy ruch po ekranie, a także zmiany jego kształtu (np. łatwo sobie wyobrazić animację kroczącej postaci).

Program w Basicu generujący kod maszynowy pozwala również na łatwe utworzenie kształtów wszystkich graczy. Po przepisaniu programu i uruchomieniu go na ekranie pojawi się ramka i pytanie o numer postaci. Cursor jest przemieszczany standardowo, klawisz „spacja” służy do zapalania i gaszenia wybranego punktu, „ESC” pozwala wybrać inną postać bez żadnych skutków, „RETURN” zaś zapisuje utworzony kształt (pod dowolnym numerem). Niepodanie numeru kształtu powoduje przejście programu do fragmentu zapisującego kod wynikowy (C — na taśmę, a D: nazwa — na dyskietkę).

Najprostsza postać programu jest podana w listingu 1 (listing 3 jest jego odpowiednikiem, zapisanym w assemblerze MAC/65). Możliwe jest dopisywanie do głównej procedury podprogramów realizujących dodatkowe funkcje. Jako przykład pokazany jest program, który pozwala na sterowanie za pomocą joysticka graczem pierwszym. Przykład ten jest zawarty w listingu 2, a jego program źródłowy w listingu 4.

I to już właściwie wszystko — wypada mi tylko życzyć przyjemnej zabawy.

Andrzej Bazik

Krzysztof Klimczak

MULTI DOS

```

RT 0 REM MULTI-BOOT -> DOS
AC 1 REM Krzysztof Klimczak
UP 2 REM Copyright (c) Bajtek
NI 3 REM
UY 10 DIM PROG*(3500):PROG*(3500)="K":GRA
PHICS 0:POKE 752,1
QG 20 Q=1:FOR T=100 TO 428:S=0:FOR I=1 TO
10:READ A:PROG*(Q,Q)=CHR$(A):S=S+A:Q=
Q+1:NEXT I:READ S1
GL 30 IF S<>S1 THEN ? :? "Blad w wierszu
nr.: ";T:STOP
ZN 35 POSITION 17,12:?">>";428-T;"<< ";
40 NEXT T
XR 50 ? :? "Dane w porzadku.":? "Wloz dys
kietke i naciśnij RETURN.":POKE 764,25
5
UZ 60 IF PEEK(764)<>12 THEN 60
UH 70 OPEN #1.8,0,"D:MULTDOS.COM":? #1;PR
OG*(1,3288 ;:CLOSE #1
PJ 80 ? :? :? "GOTOWE!!":END
PF 100 DATA 255,255,0,64,33,64,169,66,141
,49,1096
CQ 101 DATA 2,169,0,141,48,2,169,34,141,4
7,753
RZ 102 DATA 2,169,12,141,4,212,169,0,141,
198,1048
ZC 103 DATA 2,133,20,165,20,201,80,208,25
0,96,1175
BW 104 DATA 0,66,23,66,112,112,112,112,11
2,112,827
CI 105 DATA 112,71,0,67,64,6,64,23,112,11
2,631
SE 106 DATA 112,66,0,68,2,65,0,66,0,67,44
6
FU 107 DATA 59,67,0,0,0,0,0,51,45,33,255
XZ 108 DATA 50,43,13,51,47,38,52,0,0,0,29
4
AC 109 DATA 0,0,0,0,0,0,0,0,240,242,482
GS 110 DATA 229,243,229,238,244,243,0,0,0
,0,1426
OU 111 DATA 0,0,0,0,0,0,0,0,109,109
LP 112 DATA 117,108,116,100,111,115,0,0,0
,0,667
PC 113 DATA 0,0,0,68,79,68,48,114,111,103
,591
DC 114 DATA 114,97,109,0,105,115,0,119,11
4,105,878
GJ 115 DATA 116,116,101,110,0,98,121,0,43
,14,719
FF 116 DATA 43,44,41,45,35,58,33,43,0,8,3
50
SK 117 DATA 33,52,33,50,41,9,0,0,0,0,218
IW 118 DATA 0,0,43,105,101,108,99,101,0,1
7,574
LZ 119 DATA 25,24,24,0,0,8,99,9,0,51,240
EY 120 DATA 45,33,50,43,13,51,47,38,52,0,
372
LN 121 DATA 0,0,0,0,0,0,226,2,227,2,457
LG 122 DATA 0,64,255,255,0,166,237,177,32
,138,1324
ID 123 DATA 170,32,33,171,169,3,133,85,16
9,7,972
TI 124 DATA 133,84,32,162,170,162,0,169,1
75,133,1220
SS 125 DATA 208,160,97,169,96,133,209,169
,9,32,1282
VW 126 DATA 225,169,169,0,141,243,2,141,1
,212,1303
JF 127 DATA 169,255,141,252,2,173,252,2,2
01,18,1465
RZ 128 DATA 208,3,76,54,171,201,58,208,3,
76,1058
SW 129 DATA 78,172,201,21,208,235,76,105,
174,32,1302
JW 130 DATA 138,170,162,6,189,255,31,149,
159,202,1461
YQ 131 DATA 208,248,134,166,173,1,32,133,
167,230,1492
EA 132 DATA 167,208,6,169,128,133,167,208
,2,70,1258
NA 133 DATA 167,169,255,141,0,32,141,1,32
,141,1079
WC 134 DATA 4,32,169,190,141,5,32,169,6,1
41,889
HA 135 DATA 2,32,169,191,56,229,167,141,3
,32,1022
KZ 136 DATA 160,1,165,164,153,165,177,160
,5,165,1315
FK 137 DATA 165,153,165,177,160,9,165,167
,153,165,1479
JC 138 DATA 177,160,13,165,162,153,165,17
7,160,17,1349
AZ 139 DATA 165,163,153,165,177,160,25,17
3,3,32,1216
IF 140 DATA 153,165,177,165,164,160,4,153
,221,177,1539
CM 141 DATA 200,165,165,153,221,177,160,1
,165,162,1569
EW 142 DATA 24,105,6,153,221,177,165,163,
105,0,1119
AP 143 DATA 200,153,221,177,169,2,141,243
,2,141,1449
DE 144 DATA 1,212,32,33,171,32,230,170,32
,138,1051
FE 145 DATA 170,32,29,170,169,255,141,252
,2,169,1389
TA 146 DATA 2,133,84,32,162,170,169,48,13

```

```

9,175,1181
IP 255 DATA 133,208,169,16,133,209,169,9,
32,225,1303
WN 256 DATA 169,96,32,33,171,169,5,133,85
,169,1062
CC 257 DATA 10,133,84,32,162,170,162,0,16
9,175,1097
KH 258 DATA 133,208,160,179,169,80,133,20
9,169,9,1449
IU 259 DATA 32,225,169,160,255,140,252,2,
200,140,1575
QI 260 DATA 220,2,172,220,2,192,17,208,3,
76,1112
XW 261 DATA 0,166,173,252,2,201,12,208,23
9,32,1285
CP 262 DATA 64,172,32,149,170,32,162,170,
162,16,1129
KG 263 DATA 169,4,157,74,3,169,128,157,75
,3,939
VR 264 DATA 169,177,157,69,3,169,88,157,6
8,3,1060
ET 265 DATA 169,3,157,66,3,32,86,228,16,3
,763
AN 266 DATA 76,32,172,162,16,169,127,157,
73,3,987
JG 267 DATA 169,129,157,72,3,169,32,157,6
9,3,960
FG 268 DATA 169,0,157,68,3,169,7,157,66,3
,799
BA 269 DATA 32,86,228,48,46,32,64,172,169
,4,881
CH 270 DATA 133,85,169,14,133,84,32,162,1
70,162,1144
UX 271 DATA 0,160,30,169,176,133,208,169,
80,133,1258
LO 272 DATA 209,169,9,32,225,169,169,255,
141,252,1630
II 273 DATA 2,173,252,2,201,255,240,249,7
6,0,1450
CK 274 DATA 166,192,136,208,51,162,0,14,8
9,3,1021
CA 275 DATA 173,88,3,240,6,201,128,144,1,
232,1217
AV 276 DATA 232,138,24,109,89,3,205,1,32,
208,1041
QC 277 DATA 180,32,64,172,160,18,169,79,1
53,159,1186
LE 278 DATA 176,200,169,75,153,159,176,20
0,169,32,1509
LW 279 DATA 153,159,176,76,71,166,32,64,1
72,169,1238
WG 280 DATA 14,133,85,133,84,32,162,170,1
62,0,975
LF 281 DATA 160,90,169,177,133,208,169,16
114,101,992
DL 390 DATA 115,115,32,97,110,121,32,107,
101,121,951
VX 391 DATA 155,68,79,83,32,68,73,82,69,6
7,776
WP 392 DATA 84,79,82,89,155,48,48,48,48,4
8,729
FZ 393 DATA 32,66,121,116,101,115,32,105,
110,32,830
OP 394 DATA 98,117,102,102,101,114,155,79
,112,101,1081
AI 395 DATA 114,97,116,105,111,110,32,115
,116,97,1013
LR 396 DATA 116,117,115,58,32,32,32,32,15
5,69,758
VS 397 DATA 110,116,101,114,32,153,207,21
3,212,208,1466
XN 398 DATA 213,212,25,32,102,105,108,101
,115,112,1125
VP 399 DATA 101,99,44,29,156,111,114,32,3
4,35,755
DO 400 DATA 34,32,116,111,32,67,76,69,65,
82,684
AC 401 DATA 32,98,117,102,102,101,114,44,
29,156,895
EF 402 DATA 111,114,32,34,42,110,34,32,11
6,111,736
QA 403 DATA 32,68,73,82,69,67,84,79,82,89
,725
EN 404 DATA 32,111,102,32,100,114,105,118
,101,32,847
TB 405 DATA 35,110,44,29,156,111,114,32,3
4,64,729
EZ 406 DATA 34,32,116,111,32,82,85,78,32,
112,714
CE 407 DATA 114,111,103,114,97,109,44,29,
156,111,988
BM 408 DATA 114,32,34,70,110,34,32,116,11
1,32,685
OI 409 DATA 70,79,82,77,65,84,32,100,105,
115,809
NS 410 DATA 107,32,105,110,32,100,114,105
,118,101,924
TK 411 DATA 32,35,110,44,29,156,111,114,3
2,34,697
RC 412 DATA 68,34,32,102,111,114,32,68,79
,83,723
SP 413 DATA 58,155,67,58,76,79,65,68,32,6
9,727
QA 414 DATA 82,82,79,82,32,33,155,78,111,
32,766
ZQ 415 DATA 101,110,116,114,105,101,115,3
2,111,110,1015
TP 416 DATA 32,116,104,105,115,32,100,105
64,172,1344
TI 174 DATA 160,18,165,205,153,159,176,20
0,165,206,1607
PT 175 DATA 153,159,176,200,165,207,153,1
59,176,76,1624

```

```

UC 176 DATA 206,166,32,206,174,32,10,175,
32,64,1097
LI 177 DATA 172,162,16,169,68,141,240,6,1
69,254,1397
SD 178 DATA 157,66,3,169,0,157,74,3,157,7
5,861
YW 179 DATA 3,169,6,157,69,3,169,240,157,
68,1041
XW 180 DATA 3,32,86,228,16,3,76,250,167,7
6,937
QT 181 DATA 225,167,32,206,174,165,161,13
3,172,230,1665
AO 182 DATA 172,70,172,165,163,201,32,176
,36,165,1352
KL 183 DATA 162,133,168,165,163,133,169,1
69,0,133,1395
UV 184 DATA 170,169,32,133,171,160,0,177,
170,145,1327
XL 185 DATA 168,200,208,249,230,169,230,1
71,198,172,1995
XX 186 DATA 208,241,76,221,177,169,166,56
,229,172,1715
QW 187 DATA 133,173,165,172,133,174,169,0
,133,168,1420
PO 188 DATA 133,170,169,165,133,169,165,1
72,24,105,1405
SS 189 DATA 31,133,171,160,0,177,170,145,
168,200,1355
KO 190 DATA 208,249,198,169,198,171,198,1
74,208,241,2014
JN 191 DATA 165,162,133,168,165,163,133,1
69,169,0,1427
VR 192 DATA 133,170,165,173,133,171,76,10
5,168,32,1326
LO 193 DATA 10,175,169,68,141,240,6,169,5
8,141,1177
CF 194 DATA 242,6,169,46,141,244,6,169,42
,141,1206
WO 195 DATA 243,6,141,245,6,32,64,172,162
,16,1087
MU 196 DATA 169,3,157,66,3,169,6,157,74,3
,807
SL 197 DATA 169,0,157,75,3,169,6,157,69,3
,808
BB 198 DATA 169,240,157,68,3,32,86,228,16
,3,1002
ZU 199 DATA 76,250,167,32,33,171,169,0,13
3,84,1115
OF 200 DATA 169,12,133,85,32,162,170,169,
32,133,1097
WN 201 DATA 209,169,176,133,208,160,123,1
62,0,169,1509
VY 202 DATA 9,32,225,169,169,0,141,243,2,
141,1131
CB 203 DATA 1,212,169,1,133,201,169,2,133
,84,1105
CR 204 DATA 133,202,169,16,133,200,162,16
,169,0,1200
EK 205 DATA 157,68,3,157,73,3,169,6,157,6
9,862
SF 206 DATA 3,169,32,157,72,3,169,5,157,6
6,833
NT 207 DATA 3,32,86,228,16,3,76,250,167,3
2,893
UL 208 DATA 136,169,165,201,133,85,32,162
,170,169,1422
VS 209 DATA 32,133,209,169,6,133,208,160,
0,162,1212
UQ 210 DATA 0,169,9,32,225,169,198,200,20
8,192,1402
JP 211 DATA 169,16,133,200,169,21,133,201
,169,2,1213
CX 212 DATA 133,84,198,202,208,176,32,184
,169,76,1462
EN 213 DATA 253,168,173,0,6,201,48,176,1,
96,1122
WJ 214 DATA 201,58,176,251,32,162,170,169
,12,133,1364
RI 215 DATA 85,169,19,133,84,169,32,133,2
09,169,1202
IE 216 DATA 6,133,208,160,0,162,0,169,9,3
2,879
MY 217 DATA 225,169,32,184,169,104,104,76
,225,167,1455
PR 218 DATA 32,162,170,169,13,133,85,169,
21,133,1087
HO 219 DATA 84,169,32,133,209,169,176,133
,208,160,1473
EW 220 DATA 109,162,0,169,9,32,225,169,16
9,255,1299
OH 221 DATA 141,252,2,173,252,2,201,255,2
40,249,1767
JJ 222 DATA 96,157,66,3,152,157,68,3,165,
208,1075
RD 223 DATA 157,69,3,165,209,157,72,3,165
,210,1210
IM 224 DATA 157,73,3,32,86,228,96,162,48,
134,1019
NQ 225 DATA 207,134,206,134,205,230,207,1
65,207,201,1896
MC 226 DATA 58,208,14,134,207,230,206,165
,206,201,1629
FS 227 DATA 58,208,4,134,206,230,205,136,
208,231,1620
MD 228 DATA 96,169,48,162,5,149,202,202,2
08,251,1492
BL 229 DATA 164,161,134,193,165,207,24,10
5,8,133,1294
JD 230 DATA 207,201,58,144,7,56,233,10,13
3,207,1256
CJ 231 DATA 230,193,165,206,24,105,2,101,
193,133,1352
DL 232 DATA 206,134,193,201,58,144,7,56,2
33,10,1242

```


VC 233 DATA 133,206,230,193,165,205,24,10
5,1,101,1363
TS 234 DATA 193,133,205,201,58,144,19,56,
233,10,1252
NB 235 DATA 133,205,230,204,165,204,201,5
8,208,6,1614
ZB 236 DATA 169,48,133,204,230,203,136,20
8,179,162,1672
PB 237 DATA 4,181,203,157,137,176,202,16,
248,96,1420
AO 238 DATA 173,54,2,133,194,173,55,2,133
,195,1114
SA 239 DATA 169,170,141,55,2,169,160,141,
54,2,1063
ZH 240 DATA 96,165,194,141,54,2,165,195,1
41,55,1208
AF 241 DATA 2,96,104,64,169,0,133,208,133
,209,1118
PM 242 DATA 133,210,96,32,128,170,169,165
,133,106,1342
GT 243 DATA 32,162,170,169,12,162,0,32,22
5,169,1133
VF 244 DATA 169,80,141,121,7,169,1,141,68
,2,899
UP 245 DATA 32,162,170,162,0,169,175,133,
208,160,1371
JR 246 DATA 53,169,12,157,74,3,169,0,157,
75,869
SY 247 DATA 3,169,3,32,225,169,32,230,170
,76,1109
UH 248 DATA 0,166,173,48,2,133,192,173,49
,2,938
RD 249 DATA 133,193,160,2,177,192,201,66,
240,40,1404
ZV 250 DATA 160,3,162,3,177,192,136,145,1
92,200,1370
HT 251 DATA 200,202,208,246,136,169,2,145
,192,173,1673
HW 252 DATA 48,2,133,192,173,49,2,133,193
,160,1085
DD 253 DATA 39,185,57,175,153,0,165,136,1
6,247,1173
UW 254 DATA 96,160,55,32,162,170,162,0,16
3,209,1142
KP 147 DATA 169,176,133,208,160,137,162,0
,169,9,1323
US 148 DATA 32,225,169,32,162,170,169,48,
133,209,1349
XD 149 DATA 169,176,133,208,160,159,162,0
,169,9,1345
SO 150 DATA 32,225,169,169,5,133,84,32,16
2,170,1181
NE 151 DATA 169,240,133,209,169,176,133,2
08,160,181,1778
ST 152 DATA 162,0,169,9,32,225,169,32,162
,170,1130
VF 153 DATA 169,16,133,209,169,6,133,208,
160,240,1443
RZ 154 DATA 162,0,169,5,32,225,169,173,24
0,6,1181
ZU 155 DATA 201,35,208,10,173,241,6,201,1
55,208,1438
CZ 156 DATA 50,76,0,166,201,42,208,3,76,1
89,1011
WA 157 DATA 168,201,64,208,3,76,72,168,20
1,70,1231
MM 158 DATA 208,3,76,22,168,201,68,208,22
,173,1149
US 159 DATA 241,6,201,155,208,15,169,0,14
1,68,1204
PB 160 DATA 2,169,192,133,106,32,149,170,
108,10,1071
RM 161 DATA 0,32,64,172,32,149,170,162,16
,169,966
YQ 162 DATA 3,157,66,3,169,240,157,68,3,1
69,1035
UV 163 DATA 6,157,69,3,169,8,157,74,3,169
,815
JS 164 DATA 128,157,75,3,32,86,228,48,85,
169,1011
MO 165 DATA 11,157,66,3,169,0,157,68,3,16
9,803
XP 166 DATA 32,157,69,3,165,166,157,72,3,
165,989
SB 167 DATA 167,157,73,3,32,86,228,48,55,
169,1018
AY 168 DATA 11,157,66,3,169,177,157,69,3,
169,981
WG 169 DATA 0,157,73,3,169,159,157,68,3,1
69,958
LQ 170 DATA 81,157,72,3,32,86,228,48,25,3
2,764
KB 171 DATA 64,172,160,18,169,79,153,159,
176,200,1350
OZ 172 DATA 169,75,153,159,176,200,169,32
,153,159,1445
SA 173 DATA 176,76,206,166,32,251,169,32,
,115,107,931
GI 417 DATA 33,155,73,110,115,101,114,116
,32,66,915
PI 418 DATA 79,79,84,68,73,83,75,32,38,32
,643
AN 419 DATA 112,114,101,115,115,32,82,69,
84,85,909
PZ 420 DATA 82,78,155,255,255,128,191,192
,191,169,1696
UE 421 DATA 164,133,2,169,165,133,3,169,1
67,133,1238
XK 422 DATA 172,169,162,133,168,169,163,1
33,169,169,1607
GV 423 DATA 0,133,170,169,32,133,171,160,
0,177,1145
NP 424 DATA 170,145,168,200,208,249,230,1
69,230,171,1940

WN 425 DATA 198,172,208,241,165,2,133,12,
165,3,1299
GN 426 DATA 133,13,234,234,234,32,6,164,3
2,3,1085
HJ 427 DATA 2,108,10,0,255,255,224,2,225,
2,1083
VB 428 DATA 128,191,224,2,225,2,171,170,0
,0,1113
ZD 336 DATA 3,105,0,141,5,3,238,10,3,208,
716
BG 337 DATA 3,238,11,3,198,198,208,188,76
,7,1130
WI 338 DATA 172,32,33,171,169,5,133,85,16
9,10,979
AX 339 DATA 133,84,32,162,170,162,0,169,1
77,133,1222
ID 340 DATA 208,160,128,169,80,133,209,16
9,9,32,1297
SM 341 DATA 225,169,160,255,140,252,2,200
,140,220,1763
SD 342 DATA 2,172,220,2,192,17,208,3,76,0
,892
FW 343 DATA 166,173,252,2,201,12,208,239,
169,0,1422
QI 344 DATA 141,4,3,141,11,3,169,1,141,1,
615
TW 345 DATA 3,169,82,141,2,3,169,32,141,5
,747
WJ 346 DATA 3,169,1,141,10,3,32,83,228,16
,686
FM 347 DATA 3,76,32,172,173,1,32,133,198,
76,896
QI 348 DATA 34,174,32,162,170,169,32,133,
209,169,1284
DF 349 DATA 176,133,208,160,94,162,0,169,
9,32,1143
LF 350 DATA 225,169,32,162,170,169,4,133,
209,169,1442
VU 351 DATA 6,133,208,160,0,162,0,169,5,3
2,875
XT 352 DATA 225,169,173,0,6,201,89,208,8,
173,1252
EL 353 DATA 1,6,201,155,208,1,96,104,104,
76,952
WM 354 DATA 206,166,173,241,6,201,49,176,
24,160,1402
CQ 355 DATA 18,169,49,153,159,176,200,169
,54,153,1300
FV 356 DATA 159,176,200,169,48,153,159,17
6,104,104,1448
TH 357 DATA 76,206,166,201,53,176,228,173
,242,6,1527
LF 358 DATA 201,155,208,221,96,69,58,125,
155,0,1288
VR 359 DATA 0,45,53,44,52,36,47,51,0,98,4
26
MM 360 DATA 121,0,33,52,33,50,41,0,8,99,4
37
DX 361 DATA 9,0,17,25,24,24,0,51,45,33,22
8
DI 362 DATA 50,43,13,51,47,38,52,0,0,80,3
,133,209,1464
MV 282 DATA 169,9,32,225,169,76,218,171,3
2,138,1239
EL 283 DATA 170,162,16,169,12,157,66,3,32
,86,873
RX 284 DATA 228,96,32,33,171,169,1,133,85
,169,1117
CC 285 DATA 10,133,84,32,162,170,162,0,16
9,175,1097
QP 286 DATA 133,208,160,211,169,80,133,20
9,169,9,1481
IU 287 DATA 32,225,169,160,255,140,252,2,
200,140,1575
RK 288 DATA 220,2,172,220,2,192,17,208,3,
76,1112
PJ 289 DATA 0,166,173,252,2,201,12,208,23
9,169,1422
IG 290 DATA 1,141,1,3,169,82,141,2,3,169,
712
JG 291 DATA 0,141,11,3,141,4,3,169,191,14
1,804
NX 292 DATA 5,3,169,48,141,10,3,32,83,228
,722
SF 293 DATA 16,3,76,32,172,238,10,3,169,1
28,847
TJ 294 DATA 141,4,3,32,83,228,48,240,160,
240,1179
DT 295 DATA 185,0,191,132,193,69,193,208,
6,200,1377
KG 296 DATA 208,244,76,232,172,169,14,133
,84,169,1501
EZ 297 DATA 6,133,85,32,162,170,169,48,13
3,209,1147
UG 298 DATA 169,176,133,208,160,64,162,0,
169,9,1250
WX 299 DATA 32,225,169,76,218,171,32,33,1
71,32,1159
TC 300 DATA 162,170,169,2,133,84,169,10,1
33,85,1117
ZK 301 DATA 169,32,133,209,169,175,133,20
8,160,250,1638
JE 302 DATA 162,0,169,9,32,225,169,173,0,
191,1130
JB 303 DATA 208,31,169,12,133,84,169,8,13
3,85,1032
CS 304 DATA 32,162,170,169,48,133,209,169
,177,133,1402
YR 305 DATA 208,160,103,162,0,169,9,32,22
5,169,1237
AT 306 DATA 76,218,171,160,0,132,211,185,
0,191,1344
OM 307 DATA 240,7,230,211,200,192,10,208,
244,165,1707

XV 308 DATA 211,133,212,169,64,141,0,6,16
74
HN 363 DATA 114,101,115,115,58,32,34,67,3
4,32,702
RE 364 DATA 102,111,114,32,66,79,79,84,32
,67,766
QM 365 DATA 65,83,83,69,84,84,69,44,29,15
6,766
DG 366 DATA 127,31,31,31,34,68,34,32,102,
111,601
ZP 367 DATA 114,32,77,85,76,84,73,66,79,7
9,765
BV 368 DATA 84,32,102,105,108,101,115,44,
29,156,876
CY 369 DATA 127,31,31,31,34,66,34,32,102,
111,599
MY 370 DATA 114,32,66,79,79,84,68,73,83,7
5,753
YK 371 DATA 155,73,110,115,101,114,116,32
,99,97,1012
QI 372 DATA 115,115,115,101,116,116,101,3
2,38,32,881
AG 373 DATA 112,114,101,115,115,32,82,69,
84,85,909
PD 374 DATA 82,78,155,73,110,115,101,114,
116,32,976
AC 375 DATA 97,32,77,85,76,84,73,66,79,79
,748
UZ 376 DATA 84,32,100,105,115,107,32,38,3
2,112,757
EW 377 DATA 114,101,115,115,32,82,69,84,8
5,82,879
BR 378 DATA 78,155,77,85,76,84,73,66,79,7
9,852
BM 379 DATA 84,32,68,73,82,69,67,84,79,82
,720
EV 380 DATA 89,155,83,101,108,101,99,116,
32,112,996
GY 381 DATA 114,111,103,114,97,109,58,155
,84,104,1049
HH 382 DATA 105,115,32,105,115,32,110,111
,116,32,873
NX 383 DATA 97,32,66,79,79,84,32,99,97,11
5,780
ZS 384 DATA 115,101,116,116,101,32,102,10
5,108,101,997
EB 385 DATA 33,155,84,104,105,115,32,105,
115,32,880
DQ 386 DATA 110,111,116,32,97,32,77,85,76
,84,820
RB 387 DATA 73,66,79,79,84,32,100,105,115
,107,840
NI 388 DATA 33,155,29,65,114,101,32,121,1
11,117,878
XK 389 DATA 32,115,117,114,101,63,155,80,
9,32,1137
SG 309 DATA 141,1,6,141,3,6,141,24,6,169,
638
UM 310 DATA 45,141,2,6,169,155,141,28,6,1
69,862
YK 311 DATA 0,133,193,169,30,133,192,169,
5,133,1157
GO 312 DATA 84,238,0,6,164,192,162,0,165,
0,1031
OC 313 DATA 191,157,4,6,200,232,224,20,20
8,244,1486
RN 314 DATA 132,192,164,193,185,0,191,230
,193,168,1648
LA 315 DATA 32,251,169,165,205,141,25,6,1
65,206,1365
FF 316 DATA 141,26,6,165,207,141,27,6,32,
162,913
LP 317 DATA 170,169,6,133,85,169,48,133,2
09,169,1291
HZ 318 DATA 6,133,208,162,0,160,0,169,9,3
2,879
HH 319 DATA 225,169,198,212,208,181,32,16
2,170,169,1726
HU 320 DATA 11,133,85,169,17,133,84,169,1
5,133,949
LA 321 DATA 209,169,176,133,208,160,14,16
2,0,169,1400
JL 322 DATA 11,32,225,169,32,162,170,169,
255,141,1366
HY 323 DATA 252,2,169,2,141,243,2,141,1,2
12,1165
IG 324 DATA 169,17,133,84,169,27,133,85,1
69,6,992
NQ 325 DATA 133,208,169,2,133,209,160,240
,162,0,1416
KC 326 DATA 169,5,32,225,169,173,240,6,20
1,65,1285
EF 327 DATA 176,3,76,0,166,56,233,65,197,
211,1183
SB 328 DATA 176,246,168,185,0,191,133,198
,185,10,1492
BZ 329 DATA 191,141,10,3,185,20,191,141,1
1,3,896
MQ 330 DATA 169,0,141,4,3,169,32,141,5,3,
667
CN 331 DATA 32,83,228,16,34,32,33,171,32,
162,823
VX 332 DATA 170,169,14,133,85,169,10,133,
84,169,1136
CO 333 DATA 32,133,209,169,177,133,208,16
0,90,162,1473
PY 334 DATA 0,169,9,32,225,169,76,218,171
,173,1242
GE 335 DATA 4,3,24,105,128,141,4,3,173,5,

Na dyskietce zawierającej programy opublikowa-
ne w tym numerze (patrz strona 2) znajduje się —
oprócz programu „MultiDOS” w Basicu — także go-
towy program w języku maszynowym.

A może zagramy?

Poprzednią „Szkółkę” zakończyłem twierdzeniem, że pisanie programu należy rozpocząć od najprostszego możliwego schematu blokowego. Spróbujmy więc korzystając z takiego założenia napisać prostą grę. Na przykładzie tej gry prześledzimy kolejne etapy powstawania każdego programu.

Pierwszym punktem jest zawsze pomysł. Może to być potrzeba rozwiązania jakiegoś problemu z dowolnej dziedziny, np. matematycznego, fizycznego, biologicznego itd. Może chcemy drukować wizytówki lub „firmowe” nadruki na kartkach listu. Może niezbędne jest skatalogowanie biblioteki lub innych zbiorów. Takie przykłady można mnożyć bez końca i właśnie tak powstają programy użytkowe. My jednak mieliśmy zająć się grą. Czas i miejsce akcji można wybrać od $-\infty$ do $+\infty$ (znak ∞ oznacza w matematyce nieskończoność), nie będziemy się więc zastanawiać i szukać pomysłów, lecz weźmy coś z niedawnej historii.

LEGENDA

Zaczynamy od legendy. Nie od legendy w znaczeniu podania ludowego, lecz legendy, jako opisu wprowadzającego gracza w atmosferę gry. Przyda się tu odrobina zdolności literackich (nawet zaciekły komputerowiec musi być czasem niezłym polonistą).

Jest wrzesień 1939 roku. Niemieckie wojska bez wypowiedzenia wojny napadły na Polskę. Cały kraj stanął do walki. W nowoczesnej wojnie ważna jest jednak nie tylko liczba dział, czołgów, samolotów i żołnierzy, ale także — a może przede wszystkim — zaopatrzenie w żywność, lekarstwa, amunicję i paliwo. Tu jest zajęcie dla Ciebie. Jako dowódca okrętu podwodnego ORP „Orzeł” patrolujesz wody Bałtyku, aby nie dopuścić do przetransportowania tych materiałów, zwanych często krwią armii. Podczas patrolu napotykasz na nieprzyjacielski statek eskortowany przez dwa niszczyciele. Twoim zadaniem jest zniszczenie tego statku.

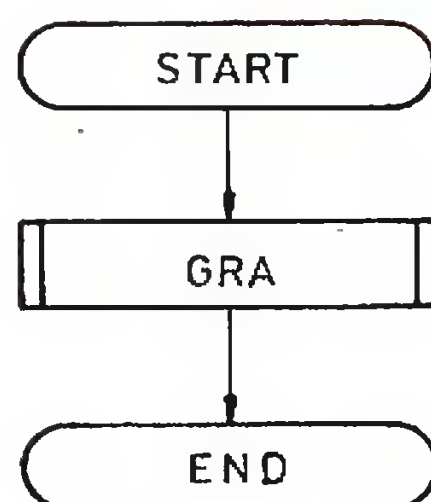
Mając temat gry przystępujemy do określenia podstawowych założeń. Część z nich została podana w legendzie, choć niekoniecznie bezpośrednio — niektórych trzeba się domyśleć. Oto one:

- gra dla jednego gracza;
- gracz steruje okrętem podwodnym;
- komputer steruje statkiem i jego eskortą;
- gracz zwycięża, gdy zatopi statek;
- komputer zwycięża, gdy zatopi okręt podwodny;
- albo
- gdy statek wejdzie do portu;
- albo
- gdy gracz zużyje wszystkie torpedy.

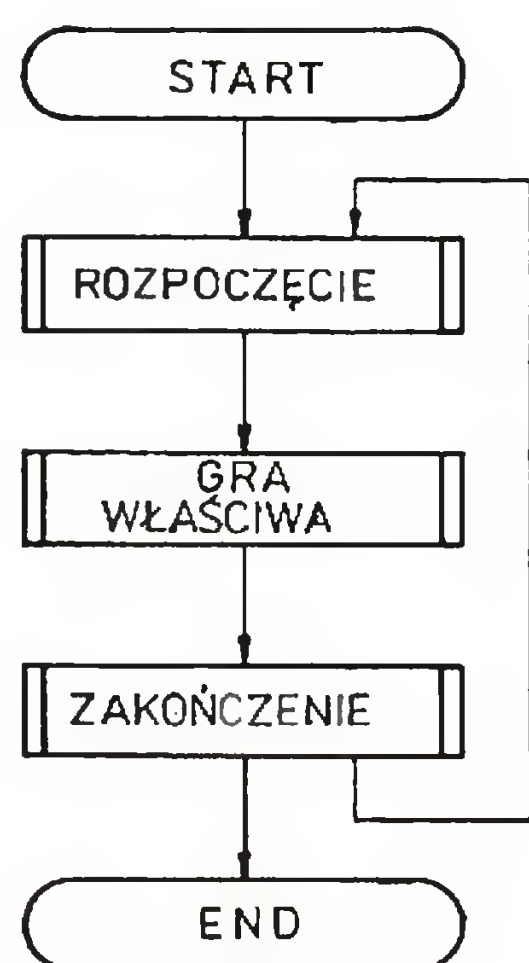
Dalsze założenia wynikają z warunków technicznych komputera (ograniczona plansza) i obiektu gry (ograniczona liczba torped na okręcie podwodnym). Dla uproszczenia pomijamy tu czas przebywania okrętu pod wodą i zapas pocisków na niszczycielach).

PIERWSZY ALGORYTM

Przerwiemy teraz ustalanie założeń i narysujemy pierwszy, wstępny schemat blokowy naszego programu. Jak widać na poniższym rysunku, jest on bardzo prosty, ale zaraz będziemy go rozbudowywać.



Zastanówmy się, z jakich elementów powinna być złożona gra. Przede wszystkim można wyróżnić trzy główne części: rozpoczęcie, właściwą grę i zakończenie. Oczywiście, po zakończeniu gry musi istnieć możliwość zagrania jeszcze raz lub całkowitego zakończenia programu. Po przeniesieniu tego na schemat otrzymamy:



Nasze dalsze postępowanie będzie polegało właśnie na takim dzieleniu każdego fragmentu programu na części. Podział kończy się po osiągnięciu takiego poziomu, który pozwala się już zapisać w wybranym języku programowania. Dla potrzeb naszej „Szkółki” wybrałem dwa języki — **Atari Basic** i **Action!**. Zanim jednak będziemy mogli coś zapisać na komputerze, czeka nas jeszcze trochę „papierkowej” roboty.

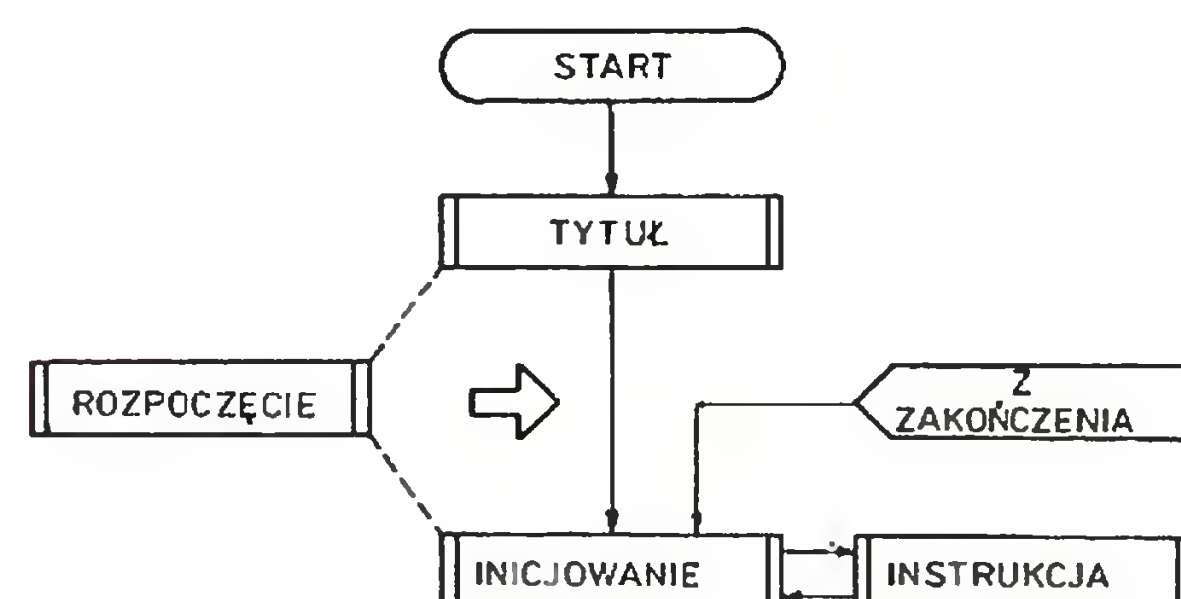
ANALIZA ELEMENTÓW

Brzmi to groźnie, lecz w rzeczywistości zaczęliśmy już realizować ten etap układania programu. Teraz trzeba go tylko pogłębić. Zaczniemy od początku, czyli od „rozpoczęcia”.

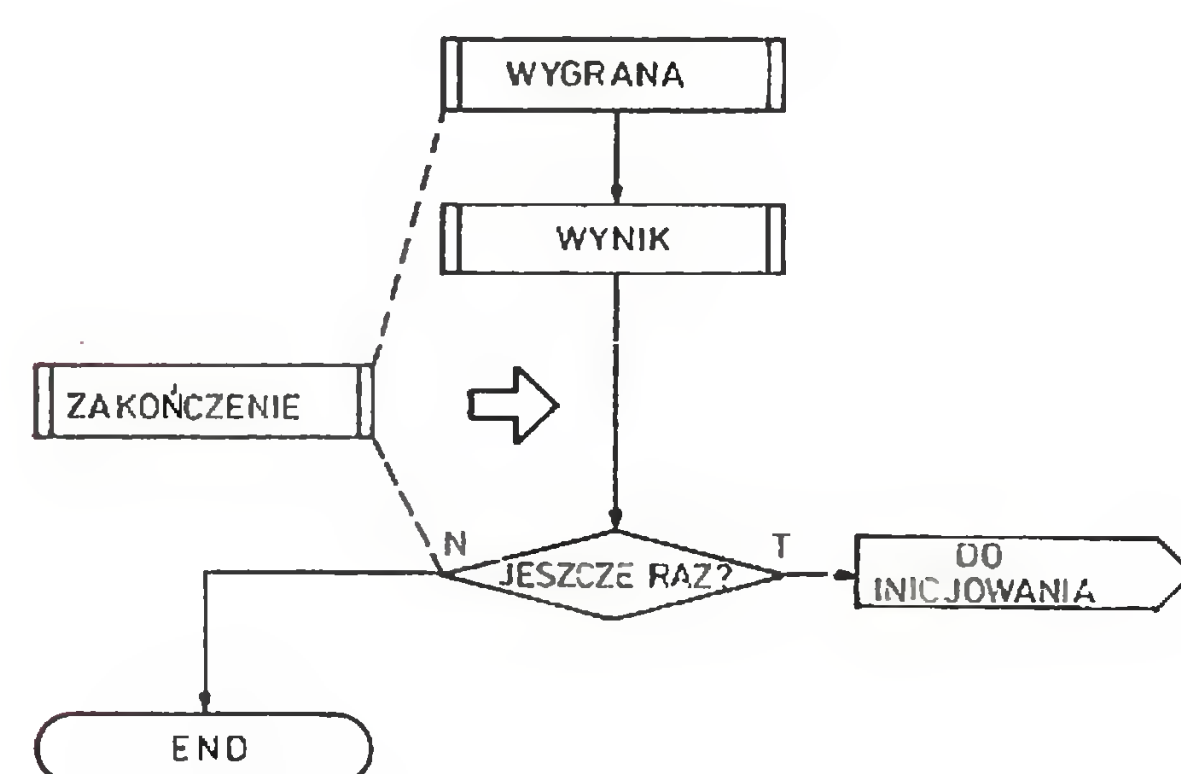
Każdy przyzwoity program powinien mieć na początku tytuł. W czasie, gdy użytkownik go ogląda, można już wykonać czynności wstępne i ustalić pewne parametry, które nie będą zmieniane podczas gry. Realizuje się tu odczytywanie stałych wartości, deklarowanie zmiennych i ewentualnie przepisywanie procedur w języku maszynowym. Ta część programu jest zawsze wykonywana tylko jeden raz.

Drugim etapem, wykonywanym na początku każdej kolejnej gry, jest inicjowanie parametrów, które

ulegają zmianie podczas programu. Można tu zaliczyć, między innymi, liczbę „żyć”, poziom trudności itp. Ponadto należy umożliwić graczowi zapoznanie się z instrukcją programu, przy czym musi istnieć możliwość pominięcia tej instrukcji. W ten sposób dzieląc na części „rozpoczęcie” uzyskujemy:



Podobnie możemy podzielić końcową część programu. Będzie się ona składać również z trzech części. Najpierw określamy zwycięzcę. Trzeba w tym przypadku sprawdzić spełnienie warunków podanych w założeniach. Sprawdzanie to jest często wykonywane jeszcze podczas gry, gdyż niekiedy łatwiejsze jest ułożenie programu. W drugiej części obliczamy wynik i ewentualny rekord (jeśli jest w grze zapamiętywany), a następnie wyświetlamy ten wynik. Ostatnim etapem jest pytanie użytkownika, czy chce ponownie zagrać. W zależności od odpowiedzi program przechodzi do inicjowania lub po prostu się kończy. W tym drugim przypadku powinny być odtworzone standardowe parametry pracy komputera. W rezultacie efektem podziału „zakończenia” jest:



Na tym przerwiemy analizowanie naszego programu, gdyż główna jego część wymaga bardziej szczegółowego opisu. Ciąg dalszy zostanie opublikowany w następnym wydaniu „Mojego Atari”.

Wojciech Zientara

Action!



PARAMETRY

Parametry pozwalają na przekazywanie wartości do procedury. Prawdopodobnie zdziwisz się, że jest to konieczne, skoro można użyć zmiennych globalnych zarówno do przekazania wartości do procedury, jak i do wykorzystania w niej. Jest to prawda, lecz istnieją co najmniej dwie przyczyny, dla których stosuje się parametry:

- umożliwia to wykorzystanie jednej procedury do różnych celów;
- pozwala to na manipulowanie wartościami zmiennych wewnątrz procedury bez zmiany wartości zmiennych globalnych.

Omówimy oba te zastosowania oddzielnie, w podanej wyżej kolejności, lecz najpierw konieczne jest podanie formatu listy parametrów.

Parametry w deklaracji PROC lub FUNC

([<dekl_zmien>] ;,<dekl_zmien>:)
gdzie <dekl_zmien> jest normalną deklaracją zmiennej, która jednak nie może zawierać określenia adresu lub wartości początkowej.

Przykłady:

```
PROC Test(BYTE chr,num,i CARD x,y)
INT FUNC DoCommand(INT cmd CARD ptr BYTE
offset)
CARD FUNC Square(BYTE x)
PROC Jump()
```

Parametry w wywołaniu PROC lub FUNC

([<wyraż_arytm>] ;,<wyraż_arytm>:)
gdzie <wyraż_arytm> jest dowolnym wyrażeniem arytmetycznym.

Przykłady:

```
Test(kot,pies,licznik,2500,$8D00)
```

```
sqr=Square(liczba)
Jump()
x=DoCommand(temp,zmiana,A)
```

UWAGA: Każda procedura może mieć maksymalnie osiem parametrów. Użycie większej liczby spowoduje błąd podczas kompilacji.

Czas teraz na trochę wyjaśnień. Poniższy przykład pokazuje, jak używać parametrów, i objaśnia pierwszą zaletę stosowania parametrów.

Pokazana poniżej funkcja sprawdza, czy zmienna "znak" typu BYTE jest małą literą. Jeżeli tak, to zwraca odpowiadającą jej dużą literę. W przeciwnym wypadku funkcja zwraca po prostu zmienną "znak". Zauważ, że nigdzie nie deklarujemy "znak". Sposób jej zadeklarowania przedyskutujemy po przykładzie.

```
BYTE FUNC maladuza()
```

```
IF znak>='a' AND znak<='z' THEN
RETURN (znak-$20) ;$20 to różnica
FI ;miedzy duzymi i malymi literami
RETURN (znak) ;w zestawie ATASCII
```


Teraz trzeba zdecydować, gdzie będzie zadeklarowana zmienna "znak". Oczywiście wiesz, że można ją zadeklarować globalnie lub lokalnie. Jeżeli zmienna ta zostanie zadeklarowana lokalnie, to jak nadać jej wartość? Widać więc, że nie można jej zadeklarować lokalnie, gdyż funkcja sama nadawałaby wartość zmiennej, a nie jest to czynność, której od funkcji oczekujemy. Chcielibyśmy, aby wywołanie funkcji "maladuza" było zbliżone do:

```
znak=maladuza ()
oraz aby funkcja badała "znak" i zwracała dużą literę, gdy to konieczne. Tak więc nie można zadeklarować zmiennej lokalnie. A deklaracja globalna? Teraz funkcja wykona żadaną operację, ponieważ "znak" w wywołaniu funkcji i "znak" w samej funkcji to ta sama zmienna globalna. Jest jednak jedna poważna wada deklarowania "znak" jako zmiennej globalnej. W każdym przypadku, gdy użyjemy funkcji "maladuza", otrzymamy wynik w zmiennej "znak". Jeżeli chcemy wykonać tę funkcję na zmiennej "symbol", musimy postąpić następująco:
```

```
znak=symbol
znak=maladuza ()
symbol=znak
```

Powoduje to znaczne skomplikowanie programu, gdy chcemy wykonać funkcję na kilkunastu różnych zmiennych. Ponadto, jeśli chcemy wykorzystać procedurę "maladuza" w innym programie, musimy pamiętać o zadeklarowaniu w nim zmiennej globalnej "znak".

A jeśli zadeklarujemy "znak" jako parametr funkcji? Spytaście "Jak...?". Właśnie tak:

```
BYTE FUNC maladuza(BYTE znak) ;<-
;deklaracja parametru
```

```
IF znak>='a AND znak<='z THEN
RETURN (znak-$20)
FI
RETURN (znak)
```

Ale w jaki sposób teraz wywołać funkcję? Bardzo łatwo. Wszystko, co trzeba zrobić, to wprowadzenie zmiennej jako parametru. Na przykład:

```
znak=maladuza (znak)
symbol=maladuza (symbol)
zmienna=maladuza ('a')
```

Zrobienie zmiennej "znak" parametrem funkcji pozwala na użycie tej funkcji do testowania dowolnych zmiennych w dowolnych programach, ponieważ "maladuza" zawsze zastępuje podaną zmienną swoją własną. Nie wykorzystuje ona żadnych zmiennych deklarowanych gdziekolwiek, a jednak pozwala na testowanie zmiennych. W ten sposób ominęliśmy pułapkę deklarowania zmiennej "znak" zarówno globalnie, jak i lokalnie. To właśnie nazwailiśmy zastosowaniem funkcji do różnych celów.

Druga zaleta stosowania parametrów jest znacznie trudniejsza do pokazania, lecz spróbuję to przedstawić tak prosto, jak tylko to możliwe (znów na przykładzie). Poniższa procedura pobiera dwie liczby typu CARD, dzieli pierwszą przez drugą i wyświetla wynik:

```
PROC dzielenie(CARD liczba,dzielnik)

liczba==/dzielnik
PrintC(liczba) ;wyswietlenie wyniku
RETURN
```

A teraz wykorzystanie procedury "dzielenie" w programie:

```
PROC glowna()

CARD licznik,liczba=[713]

FOR licznik=1 TO 10
DO
PrintC(liczba)
Print("/")
PrintC(licznik)
Print(" = ")
dzielenie(liczba,licznik)
PutE()
OD
RETURN
```

oraz jej rezultat:

```
713/1 = 713
713/2 = 356
356/3 = 118
118/4 = 29
```

```
29/5 = 5
5/6 = 0
0/7 = 0
0/8 = 0
0/9 = 0
0/10 = 0
```

Zwróć uwagę, że zmienna "numer" pozostaje stała, pomimo iż "liczba" się zmienia. Gdy procedura jest wywoływana, wartość "numer" jest przypisywana "liczbie", lecz "liczba" nie jest z powrotem przypisywana do "numer", gdy procedura jest opuszczana. Gdyby wartość "liczba" była przypisywana z powrotem do "numer", to rezultat byłby następujący:

```
713/1 = 713
713/2 = 356
713/3 = 237
713/4 = 178
713/5 = 142
713/6 = 118
713/7 = 101
713/8 = 89
713/9 = 79
713/10 = 71
```

Przekazywanie informacji poprzez parametry jest jednokierunkowe. Informacje mogą być przekazywane do procedury poprzez parametry, lecz nie mogą one być — ogólnie — przekazywane poprzez parametry na zewnątrz procedury. Jeżeli chcesz przekazać pojedynczą wartość z procedury, to zrealizuj ją w postaci funkcji. Możesz wtedy przekazać wartość z powrotem w instrukcji RETURN. Jeżeli chcesz przekazać z procedury więcej informacji, to musisz użyć zmiennych globalnych lub zastosować wskaźniki jako parametry.

Dopasowanie parametrów

Gdy wywołujesz procedurę, która ma parametry, pierwszy parametr podany w wywołaniu jest przypisywany pierwszej zmiennej z listy parametrów w deklaracji procedury, drugi do drugiej itd. Możesz przekazać mniejszą liczbę parametrów niż wymagana przez procedurę, lecz nigdy większą. Na przykład, jeżeli w deklaracji procedury jest 5 parametrów, to możesz przekazać do tej procedury od zera do pięciu parametrów. Pozwala to na tworzenie procedur, które wymagają różnej liczby parametrów, zależnie od tego, co mają wykonać. W takim przypadku najczęściej wykorzystuje się pierwszy parametr do określenia liczby dalszych parametrów przekazywanych do procedury.

Zgodność parametrów

Jeżeli wartość przekazana jako parametr i wartość oczekiwana przez procedurę są różnych typów, to kompilator zasygnalizuje błąd, gdyż **Action!** wymaga całkowitej zgodności typów parametrów. Na przykład: jeżeli przekażesz do procedury wartość CARD, gdy oczekiwana jest wartość BYTE, to młodszy bajt CARD zostanie przypisany zmiennej BYTE. Dodatkowy bajt (starszy bajt CARD) jest zbędny i kompilator nie wie, co ma z nim zrobić.

Typy parametrów

Dozwolonymi parametrami procedur są zmienne następujących typów:

- 1) zmienne typów podstawowych;
- 2) odwołania do tablic, wskaźników i rekordów;
- 3) nazwy tablic, wskaźników i rekordów.

W trzecim z wymienionych przypadków nazwa jest używana odpowiednio jako wskaźnik pierwszego elementu, wartość lub pierwsze pole w zmiennej, której dotyczy nazwa.

MODULE

Dyrektywa MODULE jest bardzo prostą instrukcją. Jej forma jest:

MODULE
Wskazuje to kompilatorowi, że chcesz zadeklarować jakieś dalsze zmienne globalne. Jest to bardzo

użyteczne, gdy piszesz duży program w częściach, z których każda ma swoje zmienne globalne. Jeżeli wpiszesz MODULE na początku każdej części, to kompilator doda wszystkie następujące po nim zmienne do tablicy zmiennych globalnych.

Program nie musi zawierać żadnej dyrektywy MODULE, ponieważ kompilator zakłada, że jedna taka instrukcja znajduje się na początku programu, niezależnie od tego, czy ją tam umieścisz, czy nie.

Deklaracje zmiennych globalnych muszą znajdować się bezpośrednio po dyrektywie MODULE lub na samym początku programu (co oznacza faktycznie umieszczenie ich po MODULE przyjętej przez kompilator).

DYREKTYWY KOMPILATORA

Dyrektywy kompilatora różnią się od zwykłych instrukcji języka tym, że są one wykonywane podczas kompilacji, a nie w czasie wykonywania programu. Instrukcje języka są realizowane po wydaniu w monitorze **Action!** polecenia RUN, a więc wtedy, gdy program przejmuje sterowanie komputerem. Natomiast dyrektywy kompilatora są realizowane, gdy wydasz w monitorze polecenie COMPILE, a więc, gdy sterowanie przejmuje kompilator.

DEFINE

Dyrektywa DEFINE jest bardzo podobna do funkcji zamiany w edytorze (<SHIFT><CONTROL><S>), lecz zamiana jest dokonywana podczas kompilacji. Aby to wyjaśnić, trzeba najpierw pokazać format:

```
DEFINE <ident>=<stała_tekst>[,<ident>=<stała_tekst>]
```

gdzie:
<ident> jest poprawnym identyfikatorem
<stała_tekst> jest poprawną stałą tekstową **Action!** (to znaczy ciągiem znaków ograniczonym przez cudzysłowy)

Dyrektywy DEFINE nie są wykorzystywane do tworzenia kodu wynikowego podczas kompilacji, lecz służą do uzyskania czytelniejszej postaci programu źródłowego. Kompilator zastępuje <ident>przez <stałą_tekst>w każdym miejscu programu, w którym zostanie napotkany <ident>. Na przykład, gdy kompilujesz program zawierający wiersz

```
DEFINE rozmiar="256"
kompilator zastępuje w każdym miejscu "rozmiar" przez "256". Pozwala to na zastosowanie wielu ciekawych rozwiązań (i powstanie dodatkowych kłopotów). Ponieważ DEFINE zastępuje każdy ciąg znaków, to możesz również zmienić słowa kluczowe! Jeżeli nie lubisz słowa CARD, możesz zamiast niego stosować słowo RYBA i na początku programu wpisać
```

```
DEFINE RYBA="CARD"
Teraz, za każdym razem, gdy podczas kompilacji kompilator napotka słowo "RYBA", pomyśli: „To w rzeczywistości oznacza słowo CARD, więc trzeba wpisać je zamiast RYBA”. Dla bliższego poznania tej dyrektywy jeszcze kilka przykładów:  
DEFINE liston="SET $49A=1"  
DEFINE begin="DO", end="OD"  
DEFINE jeden="1"
```

UWAGA: Nie zapomnij, że stała tekstowa musi być obustronnie ograniczona cudzysłowami.

Aby pokazać jeszcze dokładniej, co DEFINE robi, a czego nie robi, posłużymy się tabelką wyjaśniającą efekty działania DEFINE w różnych miejscach programu.

instrukcja	komentarz
DEFINE trzy = "3" PrintBE(trzy) ; trzy stany	dyrektywa wyświetla '3' i EOL zamienia 'trzy' na '3'

; trzysta bajtów
PrintE("trzy stany")

nie zmienia 'trzysta'
nie zmienia w cudzy-
słowach

INCLUDE

Dyrektywa INCLUDE pozwala na włączenie innego programu do programu właśnie kompilowanego. Załóżmy, że masz program o nazwie "IOBLOK.ACT", który realizuje procedury wejścia/wyjścia i chcesz użyć tych procedur w pisany programie. Wszystko, co musisz zrobić, to umieszczenie w pisany program następującego wiersza:

```
INCLUDE "D1:IOBLOK.ACT"
```

UWAGA: Specyfikacja pliku musi być umieszczona w cudzysłowach.

Powyższa dyrektywa musi być umieszczona w programie przez użyciem jakiegokolwiek procedury, która znajduje się w pliku "IOBLOK.ACT". Zauważ, że przykład zakłada, iż dyskietka z plikiem "IOBLOK.ACT" znajduje się w stacji dysków numer 1. Jeżeli w specyfikacji pliku nie podasz urządzenia, to kompilator przyjmie, że urządzeniem jest "D1:". Można włączyć plik z dowolnego urządzenia, z którego możliwy jest odczyt (np. "P:" jest nieprawidłowe). Oto kilka dalszych przykładów:

```
INCLUDE "D2:IOLIB.ACT"
```

```
INCLUDE "PROG1.DAT"
```

```
INCLUDE "C:"
```

UWAGA: Większość systemów operacyjnych wymaga, aby nazwa pliku była podana dużymi literami.

Użyteczną cechą dyrektywy INCLUDE jest możliwość umieszczenia takiej dyrektywy również w programie, który jest włączany przez INCLUDE (czyli możliwość zagnieżdżenia). **Action!** pozwala na maksymalne zagnieżdżenie do sześciu poziomów, lecz system operacyjny i urządzenia peryferyjne mają dodatkowe ograniczenia. Gdy zostanie przekroczona liczba zagnieżdżeń dozwolona przez system operacyjny, sygnalizowany jest błąd 161. Magnetofon pozwala na jeden poziom zagnieżdżenia, natomiast stacja dysków na trzy. Jeżeli w edytorze **Action!** nie ma żadnego programu, to liczba dozwolonych zagnieżdżeń jest zmniejszana o jedno.

SET

Dyrektywa SET jest wykorzystywana do modyfikowania zawartości pamięci RAM komputera. W czasie kompilacji SET wpisuje nową wartość do wskazanej komórki pamięci. W większości wypadków dyrektywa ta jest używana do zmiany wariantów pracy edytora i kompilatora przez program użytkownika. Format dyrektywy SET jest następujący:

```
SET <adres>=<wartość>
```

gdzie <adres> i <wartość> muszą być stałymi kompilatora.

Wynikiem działania dyrektywy SET jest wpisanie <wartości> do komórki pamięci <adres>. Jeżeli <wartość> jest większa niż 255, to jest ona wpisywana do komórek <adres> i <adres+1>. Jest to spowodowane tym, że 255 (\$FF) jest największą liczbą, którą można zmieścić w jednym bajcie, więc każda większa liczba wymaga dwóch bajtów do jej zapisania.

Przykłady:

```
SET $600=64      ;ustawia 64 w komórce $600
SET max=16       ;ustawia max=16
SET 1000=$FFFF  ;ustawia $FFFF w 1000 i 10001
SET $CF00=kot    ;ustawia kot w $CF00 i $CF01
DEFINE adr="$7000"
SET adr=$42
```

Ostatni przykład pokazuje definiowanie przez DEFINE stałej liczbowej użytej w SET. Ponieważ wartość z DEFINE jest w czasie kompilacji stała, to można ją zastosować w dyrektywie SET. Pamiętaj tylko o zdefiniowaniu stałej przed użyciem jej w SET.

UWAGA: Nie należy mylić działania dyrektywy SET podczas kompilacji z podobnym działaniem procedur Poke i PokeC w trakcie pracy programu.

ROZSZERZONE TYPY DANYCH

Rozszerzone typy danych czynią **Action!** językiem bardziej elastycznym niż inne dostępne dla komputerów Atari. Jak instrukcje strukturalne operujące grupami instrukcji prostych zwiększają możliwości języka **Action!**, tak rozszerzone typy danych operują grupami zmiennych typów podstawowych jeszcze bardziej powiększając możliwości języka.

Trzema rozszerzonymi typami danych w **Action!** są:

- 1) wskaźniki
- 2) tablice
- 3) rekordy

Opisujemy je teraz oddzielnie według kolejności na powyższej liście.

WSKAŹNIKI

Słowo "wskaźnik" przypomina rzecz, której używa nauczyciel, aby pokazać coś na mapie. Tak, to jest to. W **Action!** znaczenie wskaźnika jest bardzo podobne. Wskaźnik zawiera adres komórki pamięci, a więc wskazuje komórkę. Możesz zmienić wartość wskaźnika i wskazać nim inną komórkę, tak jak nauczyciel może przesunąć swój wskaźnik na inne miejsce mapy. Najważniejszą różnicą jest to, że wskaźnik nauczyciela pokazuje miasta lub kraje, a wskaźnik w **Action!** wartości BYTE, CARD lub INT.

Deklaracja wskaźnika

Format stosowany do deklarowania wskaźnika jest bardzo podobny do formatu deklaracji zmiennej typu podstawowego. Jedyną różnicą jest zaznaczenie, że chodzi o wskaźnik, a nie o zmienną typu podstawowego.

```
<typ> POINTER <ident> [= <adres>] ; <ident> [= <adres>];
```

gdzie:

<typ> jest podstawowym typem danych, na które będzie wskazywał wskaźnik;

POINTER jest słowem kluczowym stosowanym w celu zaznaczenia, że deklarowana zmienna jest wskaźnikiem;

<ident> jest nazwą zmiennej wskaźnikowej (wskaźnika);

<adres> określa komórkę pamięci, na którą początkowo wskazuje wskaźnik; musi to być stała kompilatora.

Ponieważ zmienna wskaźnikowa zawiera adres, to musi umożliwiać przechowanie wartości z zakresu od 0 do 65535 (od \$0 do \$FFFF), gdyż Atari ma tyle oddzielnych komórek pamięci. Aby to zapewnić, wskaźniki są zapisywane jako dwubajtowe liczby bez znaku (w kolejności młodszy, starszy bajt). Oznacza to, że są one zapisywane jako liczby typu CARD, lecz są interpretowane jako adresy.

Ponieważ wykorzystanie wskaźników jest pokazane dalej, to teraz tylko kilka przykładów deklaracji wskaźników:

```
BYTE POINTER wsk ;deklaruje wsk jako
                  ;wskaźnik wartości
                  ;typu BYTE
```

```
CARD POINTER adr ;deklaruje adr jako
                  ;wskaźnik wartości
                  ;typu CARD
```

```
INT POINTER ip=$8000 ;deklaruje ip jako wska-
                     ;źnik wartości
                     ;typu INT i ustawia go
                     ;na wskazywanie
                     ;adresu $8000
```

Operowanie wskaźnikami

Wskaźniki w **Action!** mogą być wykorzystane do operowania różnymi obiektami, gdyż można je łatwo ustawić na wskazywanie różnych komórek pamięci. Bardzo ułatwia to katalogowanie i tablicowanie informacji.

Poniższy program jest prostym przykładem, który pokazuje ideę działania wskaźników. Wprowa-

dzony jest w nim operator adresowy "^", który zostanie dokładniej opisany po przykładzie. Przykład:

```
PROC uzyciowskaznika()
```

```
    BYTE numer=$E0,      ;deklaruje dwie
    znak=$E1             ;zmienne typu BYTE
    BYTE POINTER bwsk ;deklaruje wskaźnik
                        ;zmiennej typu BYTE
```

```
    bwsk=@numer          ;bwsk wskazuje numer
    Print("bwsk wskazuje teraz adres ")
    PrintF("%H",bwsk) ;druk adresu numer
    PutE()
```

```
    bwsk^=255            ;wpisanie 255 do
                        ;wskazywanej komórki
```

```
    Print("numer jest teraz rowny ")
    PrintBE(numer)       ;pokazuje numer=255
```

```
    bwsk^=0              ;wpisanie 0 do numer
    Print("numer jest teraz rowny ")
    PrintBE(numer)       ;teraz numer=0
```

```
    bwsk=@znak           ;bwsk wskazuje znak
    Print("bwsk wskazuje teraz adres ")
    PrintF("%H",bwsk) ;druk adresu znak
```

```
    PutE()
```

```
    bwsk^='q             ;zmienia znak na 'q'
    Print("znak jest teraz rowny ")
    Put(znak)            ;pokazuje znak='q'
```

```
    PutE()
```

```
    bwsk^='z             ;zmienia znak na 'z'
    Print("znak jest teraz rowny ")
    Put(znak)            ;pokazuje znak='z'
```

```
    PutE()
```

```
RETURN
```

Wynik:

```
bwsk wskazuje teraz adres $E0
numer jest teraz rowny 255
numer jest teraz rowny 0
bwsk wskazuje teraz adres $E1
znak jest teraz rowny q
znak jest teraz rowny z
```

Zwróć uwagę, że używamy operatora "^", gdy chcemy umieścić wartość w miejscu wskazywanym przez wskaźnik. Tak więc wiersz "bwsk^=0" w powyższym przykładzie jest równoznaczny z "numer=0", ponieważ "bwsk" wskazuje w tym czasie na "numer". Pomimo iż nie użyliśmy tego w przykładzie, odwołania poprzez wskaźniki mogą być także stosowane w wyrażeniach arytmetycznych:

```
x=wsk^
```

Zauważ również, że poprawne jest "PrintF("%H,bwsk)". Oznacza to więc, że "bwsk" może być dostępny zarówno jako adres, jak i wartość typu CARD. Jest to bardzo pomocne przy usuwaniu błędów w programie, gdyż łatwo można sprawdzić, na co aktualnie wskazuje wskaźnik.

TABLICE

Tablice pozwalają na operowanie listami zmiennych przez umożliwienie dostępu do każdej zmiennej z listy przy użyciu tylko nazwy tablicy i indeksu. Zmienne w liście muszą być tego samego typu, a dozwolone są tylko typy podstawowe. Nazwa tablicy wskazuje listę zmiennych, indeks zaś określa konkretny element tej listy. Indeks jest po prostu liczbą, więc odwołanie brzmi: "Potrzebny jest n-ty element tablicy x", gdzie n jest indeksem, a x nazwą tablicy.

Deklaracja tablicy

Deklarowanie tablicy w **Action!** jest proste. Są jednak różne warianty, które pozwalają na zadeklarowanie różnych parametrów tablicy, w tym adresu, rozmiaru i początkowej zawartości. Ze względu na te warianty najpierw zobaczymy format w postaci skondensowanej.

```
<typ> ARRAY <wyrŁpocz> ; <wartŁpocz>:
```

gdzie:

<typ> jest podstawowym typem danych będących elementami tablicy;

ARRAY jest słowem kluczowym określającym tablicę;

<wartŁpocz> jest informacją wymaganą do zadeklarowania zmiennej jako tablicy elementów typu <typ>.

<wartŁpocz> ma format:

<ident>[(rozmiar)][=<adres> [<wartości>] <st_tekst>]
gdzie:
<ident> jest nazwą zmiennej (tablicy);
<rozmiar> jest rozmiarem tablicy i musi być stałą liczbową;
<adres> określa adres pierwszego elementu tablicy i musi być stałą kompilatora;
[<wartości>] określa początkowe wartości elementów tablicy, każda wartość musi być stałą liczbową;
<st_tekst> określa początkowe wartości elementów tablicy według ciągu znaków, przy czym pierwszy element będzie długością tego ciągu.

Teraz czas na kilka przykładów, które pomogą zrozumieć sposób deklarowania tablic.
BYTE ARRAY a,b ;deklaruje dwie tablice z elementami typu ;BYTE, bez określenia ich rozmiaru
INT ARRAY x(10) ; deklaruje tablicę INT zawierającą 10 elementów
BYTE ARRAY txt="To jest stała tekstowa" ;deklaruje tablicę ;BYTE i wypełnia ją stałą tekstową
CARD ARRAY tab=\$8000 ;deklaruje tablicę CARD, która ;rozpoczyna się od adresu \$8000 i nie ma ;określonego rozmiaru
BYTE ARRAY test=[4 7 18] ;deklaruje tablicę BYTE i ;wypełnia ją wartościami

UWAGA: Powinienes określać wymiar deklarowanej tablicy zawsze, gdy jest to możliwe, lecz są przypadki, gdy nie można lub nie trzeba tego robić:
— gdy nie wiesz, jak duża będzie tablica (np. przy definiowaniu parametrów procedury, gdy nie jest znany rozmiar tablicy przekazywanej jako parametr),
— gdy wypełniasz tablicę wartościami początkowymi (zarówno przy pomocy [<wartości>], jak i <st_tekst> i nie zamierzasz nic do niej dodawać.
Pamiętaj ponadto, że pierwszy bajt stałej tekstowej zawiera wartość określającą długość ciągu. W celu zwiększenia długości ciągu trzeba więc najpierw zmienić bajt długości, który jest zerowym elementem tablicy.

Reprezentacja wewnętrzna

Wewnętrzna reprezentacja tablic jest bardzo podobna do wskaźników. Jest tak, ponieważ nazwa tablicy jest rzeczywiście wskaźnikiem pierwszego elementu tablicy. Tablica jest po prostu ciągłą grupą pól, z których każdy zawiera jeden element tablicy. Rozmiar pola jest określony przez typ danych w tablicy i wynosi 1 bajt dla tablic typu BYTE oraz dwa bajty dla tablic CARD i INT.

Operowanie tablicami

Użycie tablic i operowanie nimi nie jest trudne, gdy tylko wiesz, jak deklarować tablice i jak się odwoływać do ich elementów. Deklarowanie tablic już zostało opisane, więc teraz zajmiemy się odwołaniami do elementów.
Przykład 1:

```
PROC testodwolan()
  BYTE x
  BYTE ARRAY cyfry(10)

  FOR x=0 TO 9 ;tablica cyfry ma 10
                ;elementow, wi c indeksy
                ;sa liczone od 0 do 9
  DO
    cyfry(x)=x+'A' ;x-ty element tablicy
                    ;uzyskuje wartosc x+'A'
    Put(cyfry(x)) ;drukuje x-ty element
                  ;jako znak
    Print(" ") ;odstep miedzy znakami
  OD
  PutE()
RETURN
```

Wynik 1:

A B C D E F G H I J

W powyższym programie znajdują się dwa odwołania do tablicy — "cyfry(x)" w instrukcji przypisania i "cyfry(x)" jako parametr procedury bibliotecznej Put. Te i wszystkie inne odwołania mają format: <ident>(<indeks>)

gdzie:
<ident> jest nazwą tablicy, do której się odwołujemy;
<indeks> jest numerem elementu w tej tablicy i jest wyrażeniem arytmetycznym.

Jak wyjaśniono w komentarzu przy instrukcji FOR, indeksy tablic nie rozpoczynają się od 1. Pierwszym elementem tablicy "koty" jest "koty(0)", a nie "koty(1)". Może się to wydawać dziwne, lecz szybko to zapamiętasz.
Przykład 2:

```
PROC zmianatablicy()

  BYTE ARRAY btab

  btab="To jest lancuch 1."
  PrintC(btab) ;btab jako liczba CARD
  Print(" ")
  PrintE(btab) ;btab jako tablica
  btab="To jest lancuch 2."
  PrintC(btab)
  Print(" ")
  PrintE(btab)
RETURN
```

Wynik 2:

```
10352 To jest lancuch 1.
10414 To jest lancuch 2.
```

KOMENTARZ: Wyraźnie widać, że zmianę uległ adres, na który wskazuje nazwa "btab". Ponowne przypisanie całej tablicy (gdy jest wykonywane przy użyciu stałych tekstowych) nie powoduje przepisania nowego ciągu znaków do obszaru pamięci zajętego przez stary, lecz tylko przypisuje tablicy nowe miejsce w pamięci. Powoduje to zmianę wartości "btab" tak, aby wskazywała początkowy adres nowego ciągu. Stary ciąg nadal pozostaje w pamięci, lecz nie jest niczym wskazywany, więc jest niedostępny.

Zwróć uwagę, że "PrintE(btab)" jest poprawne, ponieważ "btab" wskazuje na poprawną stałą tekstową, której typ jest zgodny z typem parametru wymaganego przez biblioteczną procedurę PrintE.
Przykład 3:

```
PROC rownetablice()

  BYTE ARRAY a="Oto stała tekstowa",
              btab

  btab=a
  PrintE(a)
  PrintE(btab)
RETURN
```

Wynik 3:

```
Oto stała tekstowa
Oto stała tekstowa
```

KOMENTARZ: Wszystko, co robi ten program, to pokazanie, że dwóm tablicom można przypisać tę samą wartość przez proste ustawienie wskaźnika tablicy na ten sam adres w pamięci. W powyższym przypadku jest to stała tekstowa, na którą wskazują obie nazwy tablic.

Mogłeś zauważyć, że nie zastosowaliśmy konstrukcji podobnej do:

```
BYTE ARRAY a=['T 'o ' 's 't 'a 'l 'a']
PrintE(a)
```

Taka konstrukcja nie działa prawidłowo. Pamiętaj, że stała tekstowa różni się od zwykłego ciągu, gdyż jej pierwszy bajt określa jej długość. Procedury, które oczekują stałej tekstowej, nie działają więc, gdy podasz im coś innego.

Na zakończenie prezentujemy przykładowy program, który wykorzystuje wszystkie, uprzednio opisane zastosowania tablic.

Przykład 4:

SCENARIUSZ: Masz program, który podaje tylko numer błędu, gdy użytkownik popełni omyłkę i chcesz, aby także wyświetlał on komunikat błędu. Można to zrobić przy pomocy tablic, jak w poniższym programie. Opis jego działania jest podany później.

```
PROC bledy(BYTE numbl)
;**** Ta procedura odczytuje numer
;bledu i wyswietla odpowiedni
;komunikat. Opis jej dzialania
;jest umieszczony ponizej.
```

```
  BYTE ARRAY komb1 ;komunikaty
```

```
  CARD ARRAY adr(6) ;zawiera adresy
                    ;komunikatow
```

```
  adr(0)="Niedozwolony rozkaz"
  adr(1)="Niedozwolony znak"
  adr(2)="Zla nazwa pliku"
  adr(3)="Liczba zbyt duza"
  adr(4)="Zly typ liczby"
  adr(5)="Nieznany blad"
  komb1=adr(numbl)
  Print("BLAD #")
  PrintB(numbl)
  Print(": ")
  PrintE(komb1)
  PutE()
```

```
RETURN ;koniec procedury 'bledy'
```

```
PROC glowna()
;**** Ta procedura sluzy tylko do
;wywolania procedury 'bledy' przy
;uzyciu wszystkich poprawnych
;numerow bledow, aby pokazac
;dzialanie tablicy.
```

```
  BYTE blad
```

```
  FOR blad=0 TO 5
  DO
    bledy(blad)
  OD
```

```
RETURN ;koniec procedury 'glowna'
```

Wynik:

```
BLAD #0: Niedozwolony rozkaz
```

```
BLAD #1: Niedozwolony znak
```

```
BLAD #2: Zla nazwa pliku
```

```
BLAD #3: Liczba zbyt duza
```

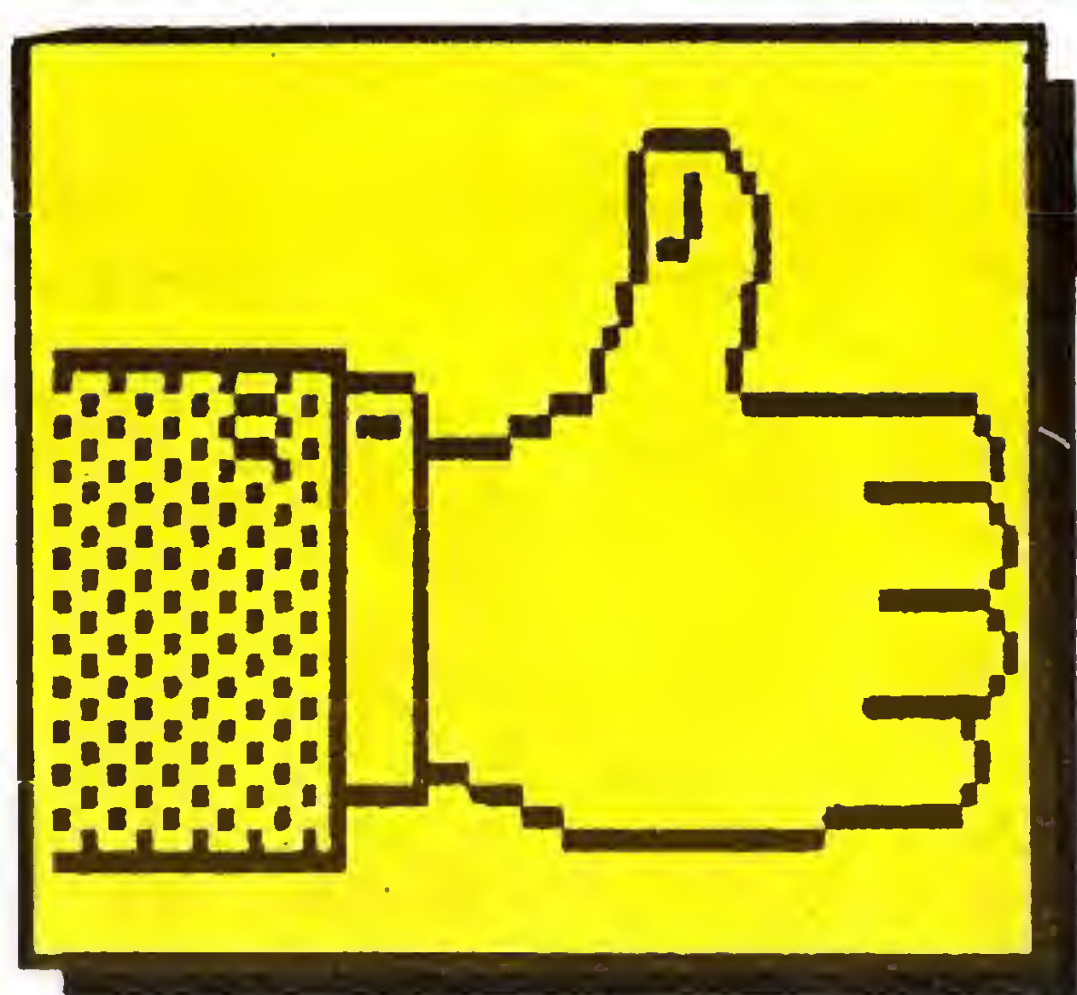
```
BLAD #4: Zly typ liczby
```

```
BLAD #5: Nieznany blad
```

KOMENTARZ: Sposób, w jaki wypełniona została tablica CARD w powyższym przykładzie, jest dziwny (jak można wypełnić tablicę CARD elementami ciągu?), lecz jest on poprawny, gdyż do elementu tablicy nie jest przypisywana sama stała tekstowa, ale jej adres. Czyni to każdy element tablicy ukrytym wskaźnikiem ciągu. Wszystkim, co trzeba uczynić, jest przypisanie wartości odpowiedniego elementu tablicy (czyli wskazującego właściwy komunikat błędu) do tablicy 'komb1', a więc uczynienie 'komb1' wskaźnikiem właściwego komunikatu. Następnie trzeba tylko wyświetlić komunikat.

Pokazany wyżej przykład jest bardzo trudny. Jego zrozumienie wymaga dokładnego zrozumienia istoty tablic i ich wewnętrznej reprezentacji. Został on jednakże umieszczony tu w celu pokazania możliwości, jakie udostępnia wykorzystanie tablic.

Wojciech Zientara



OCCENA

Mam 12 lat. Napisałem prostą grę hazardową na komputer Atari. Proszę o jej ocenę.

**Andrzej Reczuch
Bielawa**

Nadesłany program jest przedstawiony na listingu 1. Nie wiem, czy gra hazardowa jest właściwym tematem dla młodego programisty. Lepsze to jednak niż prawdziwy hazard, a program jest dobry jak każdy inny. Niestety nie można tego samego powiedzieć o jego technicznej stronie.

Budowa interpretera Atari Basic zachęca do pisania programu w takiej kolejności, w jakiej jest on wykonywany. Jednocześnie powoduje to wolniejsze wykonywanie programu, co zostało opisane w artykule „Szybki, szybszy, Atari” („Bajtek” 3/89). Drugim efektem takiego działania jest nagminne nadużywanie instrukcji GOTO.

Zajmijmy się teraz nadesłanym programem. Zgodnie z tym, co zostało wcześniej napisane, przeniesiemy początek do środka (od wiersza 1000 w drugim listingu). Na początku znajdzie się więc ten fragment programu, w którym szybkość jest rzeczywiście istotna.

Wyświetlana na początku głowa kota znajduje się z lewej strony ekranu. Aby przenieść ją na środek można dodać w każdej instrukcji PRINT odpowiednią liczbę pustych spacji. Znacznie lepszym wyjściem jest umieszczenie przed wyświetlanym tekstem przecinka, który przesunąłby wyświetlanie o 10 znaków. Podczas oczekiwania na naciśnięcie przez użytkownika klawisza <START> kot powinien mrugać. Może jest to pomyłka przy przepisywaniu, lecz mruganie to jest stanowczo zbyt szybkie, a ponadto pętla opóźniająca znajduje się tylko po pierwszej instrukcji PRINT. Ponieważ pętla ta występuje kilkakrotnie, to najlepiej napisać ją jako procedurę (wiersz 900). Plansza tytułowa zawiera jeszcze jeden niezamierzony efekt — widoczny jest na niej kursor. Likwiduje się to bardzo prosto, wystarczy dodać instrukcję POKE 752,1 (wiersz 1000).

W dalszym ciągu programu następuje wykaz „zawodników”. Ponieważ imiona startujących kotów pojawiają się w programie kilka razy, to zostały umieszczone w instrukcjach DATA, skąd są odczytywane w miarę potrzeby. Podobnie można postąpić z tytułami kotów. Zmienne tekstowe K\$ i M\$ muszą być, oczywiście, wcześniej zadeklarowane. Cały wykaz zawodników zajmuje teraz trzy wiersze programu zamiast dwunastu. Do tej fazy zostało również przeniesione ustalanie wstępnego położenia kotów na planszy.

Z uważnej analizy nadesłanego programu wyraźnie wynika, że pierwotnie w wyścigu miały brać udział tylko cztery koty. Po zwiększeniu ich liczby autorowi zaczęło brakować nazw zmiennych. A przecież wystarczy zastosować zmienne indeksowane (tablice). Upraszcza to znacznie także inne części programu, gdyż zamiast instrukcji dla każdego zawodnika oddzielnie można wykonywać tylko jedną, ale w pętli. Proszę zwrócić uwagę, jaką dało to oszczędność. Główna część była złożona z 33 wierszy i zajmowała jedną czwartą całego programu. Po zastosowaniu tablic zmieściło się to w trzech wierszach i teraz stanowi mniej niż 10 procent całości. Dodatkowo usunąłem z tej pętli rysowanie mety. Przecież wystarczy, gdy narysuje się ją jeden raz.

Dzięki zastosowaniu tablic i instrukcji DATA znacznemu skróceniu uległo również zakończenie programu. Znajduje się tu ponadto pewna ciekawostka. Po osiągnięciu mety przez któregoś kota w wierszu 110 następuje opuszczenie pętli przed jej normalnym zakończeniem. W takim przypadku niezbędne jest zastosowanie instrukcji POP. Zamiast w wierszu 110 jest ona umieszczona dopiero w wierszach 2010 i 2020. Jest to spowodowane koniecznością użycia w porównaniu (wiersz 2010) wartości I, która jest niszczone przez POP.

Podczas kursu programowania prowadzonego w cyklu „Szkółka” zwracałem uwagę na cechy, które musi posiadać dobry program. Jest tam wymieniona, między innymi, „idiotoodporność”. Proszę sprawdzić, co się stanie, jeśli użytkownik zamiast numeru kota wpisze jego imię. Taki sam efekt da naciśnięcie <RETURN> zamiast podania stawki. Takie miejsca programu muszą być zawsze zabezpieczone przy pomocy instrukcji TRAP.

Drugą ważną cechą jest „przyjazność” dla użytkownika. Należy więc dodać po każdym wyścigu informację o posiadanej sumie pieniędzy, a także pytanie o chęć dalszej gry. Również w tym miejscu została zastosowana sztuczka, która powoduje, że napis „KONIEC GRY” jest wyświetlany tylko wtedy, gdy gracz straci wszystkie pieniądze.

Poprawiony program (listing 2) wykonuje więc o kilka funkcji więcej, a pomimo tego jest szybszy i krótszy prawie o jedną trzecią. Warto dokonać dokładnego porównania obu wersji (łącznie z narysowaniem schematu). Polecam ponadto wszystkim początkującym programistom uważne czytanie „Szkółki”, która jest przeznaczona przede wszystkim właśnie dla nich.

Wojciech Zientara

LISTING 1

```
PK 1 KREDYT=5
GU 2 GRAPHICS 0
DM 10 ? :?
MY 20 ? :?
LQ 30 ? :?
TM 40 ? :?
TF 50 ? :?
NC 60 ? :?
TL 70 ? :?
RZ 80 ? :?
YM 90 ? :?
LH 100 ? :?
IH 110 ? :?
IO 120 ? :?
OK 130 ? :?
MK 140 ? :?
MW 150 ? :?
LQ 160 ? :?
OO 170 ? :?
TU 180 ? :?
HH 190 ? :?
IP 200 POSITION 2,9: ? :?
TE 210 FOR I=1 TO 5:NEXT I
CL 220 POSITION 2,9: ? :?
SA 230 IF PEEK(53279)<>6 THEN 200
SJ 400 GRAPHICS 0
IZ 410 ? :?
RG 420 ? :?
RP 430 ? :?
HH 440 ? :?
OA 450 ? :?
ZP 460 ? :?
TD 470 ? :?
SM 480 ? :?
ZI 490 ? :?
PA 500 ? :?
KR 510 ? :?
JD 520 ? :?
CD 530 ? :?
DK 600 ? :?
SR 610 INPUT X
IH 620 IF X<1 OR X>6 THEN 610
DC 630 ? :?
XQ 640 ? :?
KA 650 IF U<1 OR U>KREDYT THEN 640
XB 660 ? :?
EP 670 IF PEEK(53279)<>6 THEN 670
UH 999 GRAPHICS 0
WD 1000 A=0:B=1:C=0:D=4
OW 1010 E=0:F=7:G=0:H=10
DA 1020 AA=0:AB=13:BB=0:BA=16
IM 1030 FOR M=1 TO 70
PH 1040 SOUND 0,M,8,15
GW 1050 NEXT M
HO 1060 SOUND 0,0,0,0
KH 1070 POSITION A,B: ? :?
NL 1080 POSITION C,D: ? :?
QA 1090 POSITION E,F: ? :?
RN 1100 POSITION G,H: ? :?
UC 1110 POSITION AA,AB: ? :?
VJ 1120 POSITION BB,BA: ? :?
UR 1130 POSITION 33,0: ? :?
KT 1140 FOR P=1 TO 17
HS 1150 POSITION 34,P: ? :?
II 1160 NEXT P
HT 1170 I=INT(RND(0)*5)
IC 1180 J=INT(RND(0)*5)
IL 1190 K=INT(RND(0)*5)
HS 1200 L=INT(RND(0)*5)
OE 1210 KK=INT(RND(0)*5)
PH 1220 ZZ=INT(RND(0)*5)
AQ 1230 FOR XY=1 TO 50:NEXT XY
NZ 1240 IF A>34 THEN 2000
PN 1250 IF C>34 THEN 3000
RB 1260 IF E>34 THEN 4000
SP 1270 IF G>34 THEN 5000
SL 1280 IF AA>34 THEN 6000
UB 1290 IF BB>34 THEN 7000
KY 1300 POSITION A,B: ? :?
NN 1310 POSITION C,D: ? :?
OC 1320 POSITION E,F: ? :?
SR 1330 POSITION G,H: ? :?
XO 1340 POSITION AA,AB: ? :?
YU 1350 POSITION BB,BA: ? :?
AJ 1360 A=A+I:C=C+J
IQ 1370 E=E+K:G=G+L
DB 1380 AA=AA+XX:BB=BB+ZZ
RI 1390 GOTO 1070
ET 2000 ? :?
QT 2010 IF X=1 THEN ? :?
0
```

```
QF 2020 GOTO 9000
BC 3000 ? :?
RF 3010 IF X=2 THEN ? :?
QG 3020 GOTO 9000
IQ 4000 ? :?
RR 4010 IF X=3 THEN ? :?
QH 4020 GOTO 9000
BY 5000 ? :?
SD 5010 IF X=4 THEN ? :?
OI 5020 GOTO 9000
VO 6000 ? :?
SP 6010 IF X=5 THEN ? :?
OJ 6020 GOTO 9000
DN 7000 ? :?
TB 7010 IF X=6 THEN ? :?
RK 7020 GOTO 9000
FE 8000 KREDYT=UX5+KREDYT
AT 8010 FOR O=1 TO 5
AE 8020 FOR Y=1 TO 255
YT 8030 SOUND 0,Y,10,10
MC 8040 NEXT Y
HS 8050 SOUND 0,0,0,0
IC 8060 NEXT O
PB 8070 GOTO 400
ZP 9000 ? :?
EM 9010 SOUND 0,200,10,10
SO 9020 FOR KK=1 TO 200:NEXT KK
HN 9030 SOUND 0,0,0,0
OF 9040 KREDYT=KREDYT-U
RS 9050 FOR U=1 TO 1200:NEXT U
SH 9060 IF KREDYT<1 THEN 10000
PC 9070 GOTO 400
WI 10000 GRAPHICS 18
LI 10010 POSITION 5,4: ? :?
ZZ 10020 FOR C=1 TO 1000:NEXT C
GV 10030 RUN
```

LISTING 2

```
NC 0 REM WYSCIG KOTOW
RF 1 REM Andrzej Reczuch
UP 2 REM Copyright (C) Bajtek
NI 3 REM
IH 4 GOTO 1000
HH 100 FOR I=0 TO 5:POSITION X(I),Y(I):?
"---":NEXT I:GOSUB 900
WS 110 FOR I=0 TO 5:R(I)=INT(RND(0)*5):IF
X(I)>34 THEN 2000
KR 120 NEXT I:FOR I=0 TO 5:POSITION X(I),
Y(I):? :? :X(I)=X(I)+R(I):NEXT I:GOT
O 100
CH 900 FOR J=1 TO 50:NEXT J:RETURN
SN 999 REM
RU 1000 DIM K$(10),M$(25),X(5),Y(5),R(5):
GRAPHICS 0:POKE 752,1:POKE 82,0: ? :?
EM 1010 ? :?
PA 1020 ? :?
OS 1030 ? :?
OU 1040 ? :?
EY 1050 ? :?
DB 1060 ? :?
EF 1070 ? :?
OI 1080 ? :?
EL 1090 ? :?
HM 1100 ? :?
EO 1110 ? :?
DR 1120 ? :?
IA 1130 ? :?
BP 1140 ? :?
QM 1150 ? :?
LE 1160 ? :?
RW 1170 ? :?
RJ 1180 POSITION 10,9: ? :?
LE 1190 POSITION 10,9: ? :?
KA 1200 IF PEEK(53279)<>6 THEN 1100
MR 1210 GRAPHICS 0:POKE 82,2:POKE 752,1:R
ESTORE :?
QZ 1220 FOR I=0 TO 5:X(I)=0:Y(I)=2*I+2:R(
I)=0:NEXT I
KE 1230 FOR I=0 TO 5:READ K$,M$
TA 1240 ? :? :K$: ? :? :M$
WC 1250 NEXT I: ? :?
UF 1260 ? :?
RP 1270 TRAP 1270:INPUT X:IF X<1 OR X>6 T
HEN 1270
MT 1280 ? :?
JZ 1290 ? :?
VH 1300 TRAP 1300:INPUT U:IF U<1 OR U>KRE
DYT THEN 1300
BX 1310 ? :?
SM 1320 IF PEEK(53279)<>6 THEN 1320
OQ 1330 GRAPHICS 0:POKE 752,1
UY 1340 POSITION 33,0: ? :?
HH 1350 FOR J=1 TO 17:POSITION 34,J: ? :?
LN 1360 FOR J=1 TO 70:SOUND 0,J,8,15:NEXT
J:SOUND 0,0,0,0:GOTO 100
OR 2000 POSITION 2,19:RESTORE 5000+I*10:R
EAD K$: ? :?
MQ 2010 IF X=I+1 THEN ? :?
ZT 2020 POP:RESTORE 4990+X*10:READ K$:KR
EDYT=KREDYT-U
LC 2030 ? :?
EI 2040 SOUND 0,200,10,10:FOR I=1 TO 200:
NEXT I:SOUND 0,0,0,0
DE 2050 IF KREDYT>0 THEN 2080
YL 2060 GRAPHICS 2:SETCOLOR 2,0,0:POKE 75
2,1
GS 2070 POSITION 5,4: ? :?
HF 2080 ? :?
AU 2090 OPEN #1,4,0:"K":GET #1:I:CLOSE #
1:IF I<78 AND I>84 THEN 2090
FX 2100 IF I=78 THEN END
XA 2110 IF KREDYT>0 THEN 1210
MU 2120 RUN
OZ 3000 KREDYT=UX5+KREDYT: ? :?
FX 3010 FOR I=1 TO 5:FOR J=1 TO 255:SOUND
0,J,10,10:NEXT J:SOUND 0,0,0,0:NEXT I
:GOTO 2080
TE 5000 DATA MIAUCZUR,MISTRZ SWIATA
DL 5010 DATA FICUS,REKORDZISTA SWIATA
QX 5020 DATA MIZIAK,MISTRZ POLSKI
SJ 5030 DATA BOBUS,ZDOBYWCA PUCHARU AFRYK
I
NH 5040 DATA SMIGLY,MISTRZ EUROPY
PT 5050 DATA MINIEK,MISTRZ AMERYKI
```


ADRESY PRODUCENTÓW SPRZĘTU

Alpha Systems
4435 Maplepark Road, Stow, OH 44224, USA
interfejsy video, skanery

Anchor Automations
6624 Valjean Avenue, Van Nuys, CA 91406, USA
modemy, interfejsy

Astra Systems
5230 Clark Avenue, Lakewood, CA 90712, USA
stacje dysków

Atari, Inc.
1196 Borregas Avenue, Sunnyvale, CA 94086, USA
komputery, stacje dysków, drukarki i in.

Axlon, Inc.
1287 N. Lawrence Stat. Road, Sunnyvale, CA 94089, USA
rozszerzenia pamięci RAM

Digital Vision, Inc.
14 Oak Street, Suite 2, Needham, MA 02192, USA
interfejsy video

Duplicating Technologies, Inc.
99 Jericho Tpke., Suite 202, Jericho, NY 11753, USA
rozszerzenia stacji dysków

Epson Deutschland GmbH
Zülpicher Strasse 6, 4000 Düsseldorf 11, RFN
drukarki

Epson UK
388 High Road, Wembley, Middlesex HA9 6UH, W. Brytania
drukarki

Epson America
2780 Lomita Boulevard, Torrance, CA 90505, USA
drukarki

Fischertechnik Computing
7244 Tumlingen, Waldachtal, RFN
roboty i automaty sterowane komputerem

Happy Computing
P.O.Box 32331, San Jose, CA 95152, USA
stacje dysków i ich rozszerzenia

ICD Inc.
1220 Rock Street, Suite 310, Rockford, IL 61101, USA
interfejsy, rozszerzenia stacji dysków

Indus
9304 Deering Avenue, Chatsworth, CA 91311, USA
stacje dysków

Magna Systems
147-05 Sanford Avenue, Suite 4E, Flushing, NY 11355, USA
rozszerzenia pamięci RAM (1 MB)

Microbits Peripheral Products
225 West Third Street, Albany, OR 97321, USA
interfejsy

Mosaic Electronics
P.O.Box 708, Oregon City, OR 97405, USA
rozszerzenia pamięci RAM

Okidata
532 Fellowship Road, Mt. Laurel, NJ 08054, USA
drukarki kolorowe

Percom Data Corp.
11220 Pagemill Road, Dallas, TX 75423, USA
stacje dysków

Prometheus Products, Inc.
4545 Cushing Parkway, Fremont, CA 94538, USA
modemy

Rana Systems
21300 Superior Street, Chatsworth, CA 91311, USA
stacje dysków

Supra Corp.
1133 Commercial Way, Albany, OR 97321, USA
modemy, interfejsy, twarde dyski do XL/XE/ST, MicroNet

SWP Microcomputers Prod., Inc.
2500 E. Randol Mill Road, Arlington, TX 76011, USA
ATR800 — moduły CP/M

Zobian Controls
P.O.Box 6406, Wyomissing, PA 19610, USA
myszy do XL/XE

SPRZĘT

SPRZĘTOWY

RESET

Denerwuje Cię konieczność wyłączania i włączania komputera, gdy chcesz wczytać nowy program, lub gdy komputer zawiesił się? W prosty sposób możesz wykonać mały, opisany poniżej układ, który zrealizuje to po naciśnięciu przycisku.

Przy pomocy sprzętowego przycisku RESET możesz w każdej chwili wywołać zimny start komputera. Odpowiada to wyłączeniu komputera i ponownemu włączeniu go po kilku sekundach. Musisz tylko umieć posługiwać się lutownicą. Oczywiście, niezbędne jest także trochę wprawy, by nie uszkodzić delikatnych układów komputera. Jeśli podejmiesz się wykonania tego układu, to musisz zgromadzić następujące części:

- układ scalony 74LS123,
- dioda BAP 794 lub 1N4148,
- rezystor 4,7 kOhm,
- rezystor 150 kOhm,
- kondensator elektrolityczny 10 μ F,
- dowolny włącznik niestabilny.

Elementy te należy połączyć według zamieszczonego obok schematu. Nie podaję schematu płytki drukowanej, gdyż jest on bardzo prosty, a ponadto zależy od posiadanego modelu komputera. Zmontowany układ należy zamontować do komputera. Przewód zasilający (5 V) i masę można dołączyć do szyny równoległej, gniazda modułu (CARTRIDGE) lub gniazda joysticka (oczywiście od wewnątrz). Przyłączenie pozostałych przewodów (RESET i styk 8 MMU) zależy od modelu komputera.

600XL

Przewód RESET należy przylutować do prawej końcówki rezystora R45 (100 Ohm). Znajduje się on bezpośrednio poniżej puszk modulatora TV.

Drugi przewód — oznaczony jako „styk 8 MMU” — lutujemy do ósmego styku układu MMU. MMU jest jedynym 20-stykowym układem scalonym i ma oznaczenie U2. Znajduje się on bezpośrednio przed gniazdem modułu.

800XL

Przewód RESET należy przylutować do lewej końcówki rezystora R40. Znajduje się on z prawej strony płyty komputera, pomiędzy układami POKEY (U22) i PIA (U23).

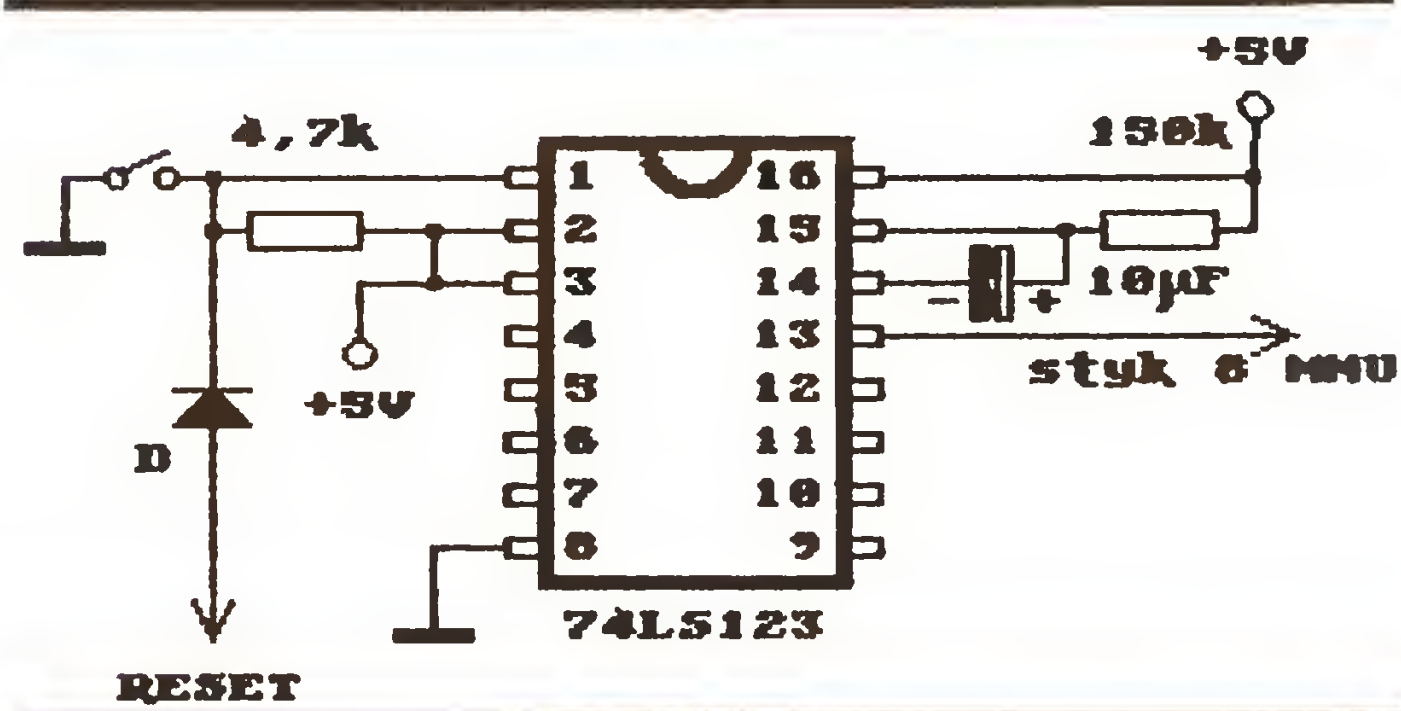
Drugi przewód — oznaczony jako „styk 8 MMU” — lutujemy do ósmego styku układu MMU. MMU jest jedynym 20-stykowym układem scalonym i ma oznaczenie U3.

65XE, 130XE i 800XE

Przewód RESET należy przylutować do górnej końcówki rezystora R40. Znajduje się on z prawej strony płyty komputera, na lewo od złącza klawiatury.

Drugi przewód — oznaczony jako „styk 8 MMU” — lutujemy do ósmego styku układu MMU. MMU jest jedynym 20-stykowym układem scalonym i ma oznaczenie U3.

Marek Zachar



— P • C • H • E • Ł • K • A —

Jedną z wad Atari Basic jest brak możliwości ładowania bloków danych.

Istniejąca instrukcja GET nie nadaje za ładowaniem danych, przy krótkich przerwach, i w dodatku wymaga stosowania pętli FOR — NEXT. Rozwiązaniem tego

problemu jest krótki, relokowalny program maszynowy. Po jego uruchomieniu procedura wpisuje się na 6 stronę (ale może to być zupełnie inne miejsce pamięci). Wymaga ona otwarcia kanału pierwszego komendą OPEN #1,4,128, „C:” a następnie wywołania rozkazem X=USR (1536, miejsce gdzie mają być wpisane dane).

```
YM 10 FOR I=1536 TO 1562:READ A:POKE I,A:
    NEXT I
NQ 20 DATA 104,104,141,85,3,104,141,84,3,
    169,255,141,88,3,141,89,3,169,5,141,82
    ,3,162,16,76,86,228
```




SOFT SYNTH

Programy muzyczne dla Atari XL/XE tworzą już liczną rodzinę, na którą składają się zarówno programy komercyjne, jak i rozpowszechniane nieodpłatnie (*public domain*). Jednym z takich programów jest Soft Snth, który wykorzystuje specjalny język programowania muzyki — MASIC.

MASIC jest językiem pozwalającym na tworzenie dźwięków, których uzyskanie trudno się nawet spodziewać w ośmiobitowym komputerze. Na przykład, podobnie jak w Amidze, jest możliwe tworzenie własnej formy dźwięku. Teoretycznie oznacza to, że można uzyskać na komputerze każdy dźwięk. Atari XL/XE nie mają jednak odpowiednich rozwiązań sprzętowych, więc trzeba to symulować programowo. Powoduje to powstanie pewnych ograniczeń.

Działający na bazie MASIC program **Soft Synth**, napisany przez Christiana Niebera, składa się z trzech niezależnych od siebie części. Sterowany z menu „Editor” pozwala na tworzenie plików zawierających dane brzmień i obwiedni. „Composer” służy do wprowadzania i redagowania muzyki w języku MASIC. Wykonanie tej muzyki umożliwia interpreter, który został połączony w integralną całość z „Composerem”.

Po wczytaniu **Soft Synth** ukazuje się na ekranie główne menu. Wymienione w nim warianty wybiera się przez naciśnięcie klawisza z pierwszą literą nazwy wariantu. Możliwe jest wybranie „Composer”, „Editor”, „DOS-u” i „Basica”. Dwa ostatnie warianty powodują odpowiednie przejście do DOS-u lub do Basica i przerywają pracę **Soft Synth**. Ponadto wariant „Optionen” umożliwia ustalenie parametrów pracy programu: dźwięk sygnalizujący odczyt i zapis (*Ladegerausch*), dźwięk klawiatury (*Tastenklick*) i delikatny przesuw obrazu (*Softscrolling*).

COMPOSER

„Composer” jest edytorem zorientowanym wierszowo, jak normalny edytor Atari Basic. Oznacza to, że program pisze się w numerowanych wierszach. Dozwolone są przy tym numery wierszy z zakresu od 1 do 32767. W każdym wierszu można

umieścić tylko jedną instrukcję. Wiersze zawierające błąd są wyświetlane ponownie ze wskazaniem miejsca wystąpienia błędu. Podczas wprowadzania instrukcji można opuszczać spacje.

Poniżej są opisane wszystkie instrukcje redakcyjne rozpoznawane przez „Composer”. Litery n i m oznaczają liczby naturalne, dozwolone zaś skróty nazw instrukcji są podane w nawiasach. W razie pominięcia parametrów instrukcji (tam gdzie jest to dozwolone) program przyjmuje parametry standardowe.

LIST (L.)

Składnia: **LIST #Device: Fname. Ext,n,m**

Przykłady:

LIST

LIST 100

LIST #D:MUZYKA.LST,100,200

Instrukcja ta powoduje zapisanie na urządzeniu *Device* fragmentu znajdującego się w pamięci programu od wiersza n do m (włącznie). Standardowymi parametrami są: *Device*=E:, n=1 i m=32767. Instrukcja LIST n powoduje zapisanie tylko jednego wiersza, a instrukcja LIST n, daje listing od wiersza n do końca programu. Za pomocą LIST można zapisać program w formie pozwalającej na późniejsze dołączenie go do innego programu.

SAVE

Składnia: **SAVE #Device:Fname.Ext**

Przykład:

Save #D:MUZYKA.SYN

Zapisuje znajdujący się w pamięci program muzyczny we wskazanym pliku dyskowym. Program jest zapisywany w postaci stokenizowanej, czyli w takiej, w jakiej jest przechowywany w pamięci.

LOAD

Składnia: **LOAD #Device:Fname.Ext,n,m**

Przykład:

LOAD #D:MUZYKA.SYN

Odczytuje do pamięci program muzyczny ze wskazanego pliku dyskowego. Program musi być zapisany w postaci stokenizowanej, czyli instrukcją SAVE.

NEW

Składnia: **NEW**

Instrukcja ta usuwa z pamięci znajdujący się tam program (podobnie jak w Atari Basic).

PLAY (P)

Składnia: **PLAY #Device:Fname.Ext,device,n**

Przykłady:

PLAY
PLAY 100
PLAY #D:MUZYKA.SYN
PLAY #D:MUZYKA.SYN,D2:,100

Podobnie jak instrukcja RUN w Atari Basic, także PLAY uruchamia wskazany program, czyli powoduje wykonanie zapisanego w nim utworu. Przerwanie utworu jest możliwe przez naciśnięcie klawisza <BREAK>. Program jest odczytywany ze wskazanego pliku dyskowego, a jeśli plik nie został podany, to uruchamiany jest program znajdujący się w pamięci komputera. Podanie numeru wiersza powoduje rozpoczęcie wykonywania programu od tego wiersza. Druga nazwa urządzenia (bez znaku #) wskazuje, skąd mają być pobierane dla wykonywanego programu dane brzmień i obwiedni.

EXIT

Składnia: **EXIT**

Powoduje opuszczenie „Composera” i odczytanie z dyskietki głównego menu. Jeżeli w stacji numer 1 nie ma dyskietki **Soft Synth**, to wyświetlany jest ostrzegawczy komunikat: „BITTE MASTERDISK EINLEGEN, RETURN”.

EDIT

Składnia: **EDIT**

Powoduje opuszczenie „Composera” i odczytanie z dyskietki „Editora”. Jeżeli w stacji numer 1 nie ma dyskietki **Soft Synth**, to wyświetlany jest ostrzegawczy komunikat: „BITTE MASTERDISK EINLEGEN, RETURN”.

ENTER (E.)

Składnia: **ENTER #Device:Fname.Ext**

Przykład:

ENTER #D:MUZYKA.LST

Odczytuje z pliku dyskowego wiersze programu zapisane instrukcją LIST. Program znajdujący się w pamięci nie jest kasowany. Instrukcja ta służy przede wszystkim do łączenia różnych kompozycji lub ich części.

DIR

Składnia: **DIR Fspec**

Przykłady:

DIR
DIR D2:*. *
DIR *.S

Odczytuje z dyskietki katalog plików określonych przez *Fspec*. Brak tego parametru powoduje odczyt całego katalogu ze stacji numer 1.

NUM

Składnia: **NUM n,m**

Przykłady:

NUM
NUM 100,2
NUM 1000

Uruchamia automatyczne numerowanie wierszy wpisywanego programu. Pierwszy wiersz otrzymuje numer równy pierwszemu parametrowi, kolejne zaś numery wierszy są zwiększane o wartość drugiego parametru. Jeśli brak drugiego parametru, to przyjmowana jest wartość 10. Tryb automatycznego numerowania jest przerywany samoczynnie po napotkaniu istniejącego już wiersza o takim numerze, jaki powinien być właśnie nadany nowemu wierszowi.

SIZE

Składnia: **SIZE**

Instrukcja ta powoduje wyświetlenie liczby określającej w bajtach wielkość wolnej pamięci, pozostałej jeszcze do dyspozycji użytkownika.

REN

Składnia: **REN n,m**

Przykłady:

REN
REN 5
REN 1000,10

Ta instrukcja służy do przenumerowania wierszy

programu znajdującego się w pamięci. Pierwszy parametr określa numer pierwszego wiersza programu, a drugi wartość, o którą są zwiększane numery kolejnych wierszy. Standardową wartością obu parametrów jest 10.

TP

Składnia: **TP liczba półtonów**

Przykłady:

TP 6
TP 0.7
TP —12

Transponuje znajdujący się w pamięci utwór o podaną liczbę półtonów. Dozwolone są także liczby ujemne i ułamki dziesiętne. Instrukcja ta nadaje się szczególnie do transpozycji utworu na inny instrument. Jedynie glissanda nie są zmieniane prawidłowo.

REP

Składnia: **REP brzmienie,brzmienie**

lub **REP obwiednia,obwiednia**

Przykłady:

REP SINUS,RECHTECK
REP PERCUS,GEIGE

Zastępuje brzmienie lub obwiednię podane jako pierwszy parametr na podane jako drugi parametr. Gdy instrukcja ta jest często stosowana, może być konieczne ponowne odczytanie wszystkich danych brzmień i obwiedni za pomocą instrukcji PLAY! Instrukcja PLAY! lub P! może być wykorzystywana tak jak PLAY.

Możliwe jest umieszczanie kilku instrukcji redakcyjnych w jednym wierszu. Trzeba je tylko rozdzielić znakiem dzielenia (/). Są one w takim przypadku wykonywane kolejno jedna po drugiej. Na przykład:

LOAD #D:MUSIK.SYN/REP SINUS,DREIECK/TP —6/PLAY

Instrukcje redakcyjne „Composera” można również umieszczać w tekście programu. Pomiędzy numerem wiersza i instrukcją trzeba wpisać znak dzielenia. Gdy interpreter napotka taką instrukcję, to przekazuje sterowanie do edytora w celu ich wykonania. Ta funkcja otwiera szerokie możliwości. Można, na przykład, wywoływać jeden utwór z wnętrza drugiego lub realizować bez końca transpozycję utworu i jego ponowne odtwarzanie. Pokazują to poniższe przykłady:

**100 /PLAY #D:MUZYKA2.SYN
100 /TP —4/PLAY**

Oprócz instrukcji redakcyjnych interpreter „Composera” rozpoznaje szereg instrukcji dźwiękowych. Można je umieszczać tylko w tekście programu, a więc nie są one wykonywane w trybie bezpośrednim (bez numeru wiersza). Instrukcje te wymagają dla określenia tworzonego dźwięku podania kilku parametrów. Najpierw zajmiemy się więc tymi parametrami.

Kanał dźwiękowy

Określa numer jednego z czterech niezależnych generatorów dźwięku istniejących w Atari.

Częstotliwość (nuta)

Teoretycznie **Soft Synth** pozwala na wybranie częstotliwości dźwięku z zakresu od 0 do 4000 Hz i z dokładnością do 0,15 Hz. Nuty z tonacji C-dur mają po prostu swoje nazwy (C, D, E, F, G, A, H). Półtony oznaczane są przez dodanie znaku „#” (C#, D#, F#, G#, A#). Chcąc otrzymać nutę spoza podstawowej oktawy (zdefiniowanej przez C=216 Hz), należy do nuty dopisać liczbę wskazującą odległość od tonacji podstawowej. C1 oznacza więc jedną oktawę wyżej od podstawowej, C-3 zaś trzy oktawy niżej. Dozwolone są wartości od —9 do 4.

Możliwe jest także bezpośrednie określenie częstotliwości dźwięku przez podanie jej wartości poprzedzonej apostrofem (np. '100 dla 100 Hz). Jeżeli podana częstotliwość odpowiada jakiejś nucie, to w listingu zostanie umieszczona ta nuta zamiast wartości bezpośredniej. Ze względu na sposób zapisu wartości w komputerze maksymalna dokładność wprowadzanej częstotliwości nie może być większa niż 1/10 Hz.

Brzmienie i obwiednia

Dla tych dwóch parametrów trzeba podać nazwy, które odpowiadają nazwom plików utworzonych przez „Editor” i zawierających dane brzmień i obwiedni.

Głośność

Ten parametr może przyjmować wartości całkowite z zakresu od 0 do 15.

Czas trwania dźwięku

Określany jest w jednostkach równych 1/50 sekundy i może wynosić od 1 do 16000 takich jedno-

stek. Przy dużych wartościach parametru występują jednak pewne odchylenia odżądanego czasu.

Czas oczekiwania na następny dźwięk

Mierzony jest w takich samych jednostkach, a dozwolone są wartości z zakresu od 0 do 65535. Czas ten nie ma żadnego związku z czasem trwania dźwięku. Przykładowo, jeśli jest on krótszy niż czas trwania dźwięku, można wywołać nowy dźwięk w innym kanale, podczas gdy ten pierwszy jeszcze brzmi.

W celu określenia dźwięku, który ma być utworzony, należy wpisać wymienione parametry w odpowiedniej kolejności, oddzielając je od siebie przecinkami. Składnia instrukcji tworzącej dźwięk wygląda więc następująco:

(numer wiersza) (kanał), (nuta), (brzmienie), (obwiednia), (głośność), (czas trwania), (czas oczekiwania)

Na przykład:

10 1,C1, SINUS,PERCUS,15,110,150

co oznacza, że wiersz 10 poleca tworzenie przez generator 1 dźwięku odpowiadającego nucie C, z brzmieniem SINUS i obwiednią PERCUS, o głośności 15 (maksymalnej), trwającego 100 jednostek (czyli 2 sekundy), a następny dźwięk zostanie utworzony po upływie 150 jednostek (3 sekundy) od jego początku.

Wpisywanie wszystkich parametrów dla każdej nuty jest bardzo męczące. Można jednak ustalić niektóre parametry za pomocą instrukcji SET.

SET

Składnia: **SET (kanał), (brzmienie), (obwiednia), (głośność), (czas trwania), (czas oczekiwania)**

Przykład:

SET 1, SINUS,PERCUS,15,100,150

W ten sposób wszystkie parametry dźwięku zostają wstępnie ustalone. Instrukcja tworzenia dźwięku może teraz wyglądać następująco:

20 1, C1

Jest również możliwe pominięcie numeru generatora (kanału). Powyższy wiersz będzie teraz wyglądał następująco:

20 C1

W takim przypadku przyjmowany jest numer generatora, użyty w poprzedniej instrukcji tworzenia dźwięku. Gdy takiej instrukcji nie ma, przyjmowany jest kanał 1. Przy listowaniu programu brakujące numery kanałów są jednak dopisywane automatycznie.

Instrukcja SET może być, oczywiście, umieszczana w programie wielokrotnie. Jest także możliwe określenie dla pojedynczego dźwięku parametrów odmiennych niż ustalone. Na przykład:

30 C1,RECHTECK (brzmienie)
30 C1,„GEIGE (obwiednia)
30 C1,RECHTECK,GEIGE (brzmienie i obwiednia)
30 C1,15,180,200 (głośność, czas trwania i czas oczekiwania)
30 C1,180,200 (czas trwania i czas oczekiwania)
30 C1,200 (czas oczekiwania)

Naturalnie trzy pierwsze i trzy ostatnie przykłady mogą być ze sobą łączone w dowolnych kombinacjach. Trzeba jedynie pamiętać, aby parametry brzmienia i obwiedni znajdowały się zawsze przed wartościami liczbowymi. Gdy została podana tylko jedna wartość liczbowa, jest ona interpretowana jako czas trwania i czas oczekiwania, trzy zaś — jako głośność, czas trwania i czas oczekiwania.

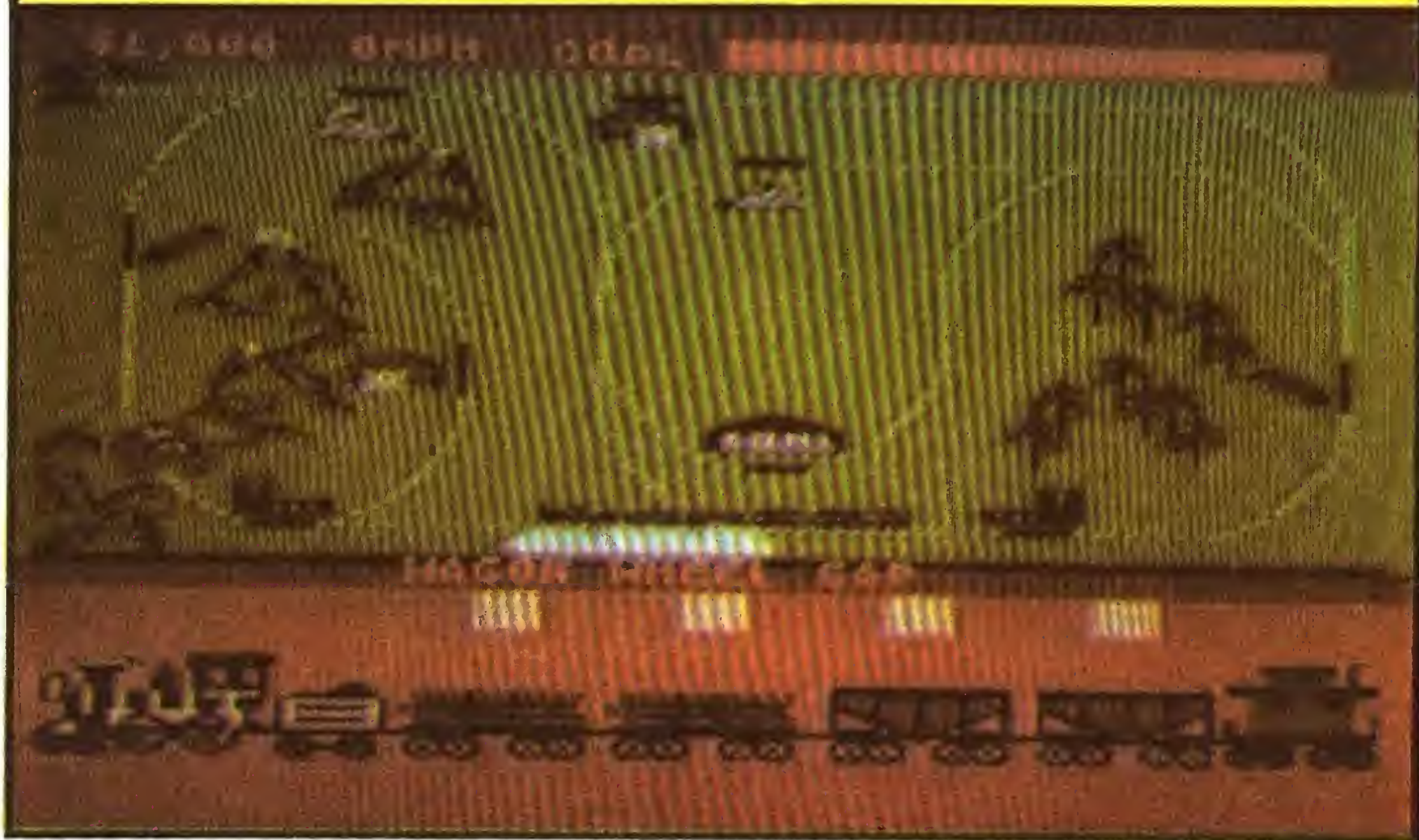
Podane tu informacje umożliwiają już tworzenie prostych melodii. W następnym numerze „Mojego Atari” zostaną opisane pozostałe instrukcje „Composera”, które pozwalają na uzyskanie różnych ciekawych efektów.

Marek Zachar

Na drugiej stronie dyskietki zawierającej programy opublikowane w tym numerze (patrz strona 2) znajduje się także program „Soft Synth”. Ponieważ jest to program „public domain”, to dodajemy go jako premię dla tych czytelników, którzy zakupią naszą dyskietkę.

TRAINS

XL/XE



Już dosyć stara, bo powstała w 1983 roku, gra „Trains” nie jest w Polsce popularna. Warto się nią jednak zainteresować, gdyż jest to jedna z licznych gier, które pokazują, jak ciężko trzeba pracować, aby zarobić kilka dolarów.

W tym przypadku gracz prowadzi poważny interes — usługi transportowe. Na początku otrzymuje 1000 dolarów oraz pociąg złożony z parowozu i pięciu wagonów. Cztery z nich służą do przewożenia ładunku, a piąty jest wagonem służbowym. Rodzaj wagonów jest zależny od rodzaju obsługiwanych zakładów (patrz tabela). Jeżdżąc po znajdujących się na planszy torach należy przewozić towar od producenta do konsumenta. Otrzymuje się za to zapłatę, lecz nie wszystko stanowi zysk. Jak w prawdziwym interesie, tak i tu są pewne koszty: podatki, opłata za surowce (węgiel dla parowozu) oraz koszty nadzwyczajne. W tej ostatniej grupie mieszczą się opłaty za naprawy uszkodzeń powstałych w wyniku katastrof oraz za nadzwyczajne dostawy węgla, jeśli zabraknie go gdzieś „w polu”.

Gra ma osiem poziomów trudności, które różnią

się rodzajem i liczbą obsługiwanych klientów oraz zestawem wagonów w pociągu. W każdym przypadku ładunki są przewożone specjalnym wagonem i tylko od określonego producenta do określonego odbiorcy. Cysterny służą do transportu ropy (OIL) z szybów wydobywczych (OILWELL) do rafinerii (RAFINERY), platformami wozi się rudę (ORE) z kopalń (ORE MINE) do fabryk (FACTORY), pnie drzew (LUMBER) z miejsc wyrębu (LUMBER CAMP) do tartaków (SAWMILL) są przewożone platformami, natomiast żywność (FRESH PRODUCE) z farm (FARM) do sklepów (MARKET) wagonami krytymi. Zestawienie zakładów i wagonów na poszczególnych planszach zawiera poniższa tabela.

poziom	1	2	3	4	5	6	7	8
szyb naft.	2	-	-	1	1	-	-	1
rafineria	2	-	-	2	1	-	-	1
wyręb	-	1	2	-	1	2	1	1
tartak	-	1	1	-	1	2	1	1
kopalnia	-	1	-	-	1	2	1	1
fabryka	-	1	-	-	1	2	1	1

farma	-	-	2	1	-	-	2	1
sklep	-	-	1	2	-	-	2	1
cysterna	4	-	-	2	2	-	-	1
platforma	-	2	2	-	1	2	1	1
węglarka	-	2	-	-	1	2	1	1
wagon kryty	-	-	2	2	-	-	2	1

Zasady gry są bardzo proste — trzeba zarabiać. Należy więc wozić towary między zakładami, zwracając uwagę na to, aby nie zużyć całego zapasu węgla. W miarę osiągania zysku odpowiedniego dla danego poziomu budowany jest nowy tor, który prowadzi na następny poziom. Po zbudowaniu tego toru wybór momentu, w którym przechodzi się na następny poziom, jest uzależniony od gracza. Szczególną uwagę należy również zwracać na to, aby u producentów nie tworzyły się zbyt duże zapasy (komunikat „...HAS SURPLUS OF...” — „...ma nadmiar...”), a u odbiorców braki („...NEEDS MORE...” — „...potrzebuję więcej...”). W krańcowym wypadku powoduje to wstrzymanie pracy („WORK STOPPAGE AT...” w danym zakładzie).

Dodatkowe komunikaty informują gracza o niewielkim zapasie węgla („TRAIN IS LOW ON COAL”) oraz o konieczności zapłacenia 200 dolarów podatku („PAYROLL TIME AGAIN”). Przy złym prowadzeniu interesu wkrótce można zobaczyć nieprzyjemną informację „YOU ARE OUT OF BUSINESS” (straciłeś firmę).

Podczas gry ekran jest podzielony na kilka części. Na samej górze podany jest aktualny stan konta i posiadany zapas węgla. Pod nimi znajduje się właściwa plansza z układem torów, zakładami i miejscem pobierania węgla (COAL). Komunikaty wyświetlane są w pasie oddzielającym planszę od wykazu wagonów i zakładów. Wykaz ten pokazuje przede wszystkim stan zapasów u wszystkich klientów albo stopień załadunku wagonów.

Program jest obsługiwany za pomocą joysticka. Ruch do przodu powoduje przyspieszanie pociągu, do tyłu zaś jego zwalnianie, a następnie — po zatrzymaniu — ruch wstecz. Zwrotnice są ustawiane ruchami joysticka na boki. Po zatrzymaniu odpowiedniego wagonu na wskazanym miejscu naciśnięcie przycisku powoduje jego załadunek lub rozładunek. Jeśli pociąg nie stoi prawidłowo, naciśnięcie przycisku wywołuje gwizd parowozu. Ponadto w grze używany jest klawisz spacji, który przełącza w najniższej części ekranu widok pociągu na widok zakładów i odwrotnie.

(ziew)

Ocena (w skali od 1 do 10):

GRAFIKA	6	OBSŁUGA	8
DŹWIĘK	4	SENS GRY	8

Producent: Spinnaker Software Corp.

Autor: Interactive Picture Software (grupa autorów)

ARCTIC ANTICS /SPY VS SPY III/

XL/XE



Znani prawie wszystkim miłośnikom gier bohaterowie programów „Spy vs. Spy” i „The Island Caper” spotkali się ponownie. Tym razem miejscem ich działania jest arktyczna wyspa, której kształt i wielkość gracz może sam wybrać.

Zadaniem obu szpiegów jest odnalezienie na pokrytej śniegiem wyspie ekwipunku złożonego z trzech przedmiotów. Mając komplet wyposażenia, należy odnaleźć ukrytą w głębokim szybie rakietę i odlecieć w kosmos. Niestety, na wyspie znajduje

się tylko jeden zestaw poszukiwanych przedmiotów i tylko jedna rakietą. Trzeba więc stoczyć walkę z konkurentem.

Każdy szpieg posiada kilka przedmiotów, które mają mu pomóc w walce i poszukiwaniach. Są to:

1. szufla — do kopania dołków;
2. piła — do wycinania zapadni w lodzie;
3. garnek z wodą — do robienia ślizgawek;
4. kilof — do przygotowania lawin;
5. rakiety śnieżne — do chodzenia po śniegu;
6. mina — są groźne tylko dla przeciwnika;
7. mapa — ułatwia orientację i poszukiwania.

Większość z tych rzeczy, poza mapą i rakietami, służy do wykonywania pułapek. Ponadto na wyspie można trafić na pułapki naturalne. Są to miejsca pokryte cienką taflą lodu oraz obszary sypkiego śniegu. Po sypkim śniegu można przejść przy użyciu śnieżnych rakiet, lecz cienki lód zawsze zapada się pod śniegiem. Każdy wypadek powoduje obniżenie temperatury ciała, którą wskazuje termometr z lewej strony ekranu. Gdy temperatura spadnie do zera, szpieg zamarza i na tym gra się dla niego kończy. Na szczęście na wyspie stoi igloo, w którym można się ogrzać (szczególnie szybko, jeśli w igloo jest ognisko). Najśmieszniej wygląda jednak bezpośrednie spotkanie obu szpiegów. W celu obniżenia temperatury ciała przeciwnika obrzucają się oni... śnieżkami.

W pierwszej grze z tego cyklu „Spy vs. Spy” — szpiegzy opuszczają ambasadę samolotami. Docierają nimi na wulkaniczną wyspę (gra „The Island Caper”). Z kolei okrętem podwodnym dostają się do Arktyki. W „Arctic Antics” zwycięski szpieg odlatuje w kosmos rakietą z napisem „Spies in Space” („Szpiegzy w Kosmosie”). Taki właśnie tytuł będzie miała czwarta część, która z pewnością wkrótce się pojawi w Polsce

(ziew)

Ocena (w skali od 1 do 10):

GRAFIKA	8	SENS GRY	9
DŹWIĘK	8		

Producent: First Star Software

Autor: Jim Nangano

FEUD

XL/XE



Zaczęło się jak w bajce...

Dawno, dawno temu, w odległych czasach, żyło sobie w małej wiosce dwóch braci czarowników, Learic i Leanoric. Z początku byli bardzo zgodnym rodzeństwem, ale pewnego dnia doszło między nimi do ogromnej sprzeczki. Różne były zdania na temat przedmiotu sporu i dziś pewnie już nie dojdziemy prawdy. Faktem natomiast pozostaje to, że bracia rozstali się i każdy zamieszkał we własnym domu. Od tej pory przyświecała im tylko jedna idea: pokonać swego krewniaka.

Jako gracz wcielasz się w postać Learica. Aby osiągnąć swój cel, musisz pozbierać zioła i zdobyć przeróżne przedmioty, które pomogą ci sporządzić mikstury pozwalające używać zaklęć i czarów.

W górnej części ekranu znajduje się widok tego frag-

mentu wioski, gdzie aktualnie przebywasz. Niewielkie kwiatki rosnące na ścieżkach to zioła i przedmioty potrzebne do przyrządzania czarodziejskich mikstur. Aby je podnieść, należy podejść do nich od dołu. Znajdująca się po lewej stronie u dołu Księga Czarów otworzy się na odpowiedniej stronie, a nazwa znalezionej rzeczy lub zioła zmieni barwę na czerwoną. Księga Czarów jest spisem wszystkich czarów. Aby ją przejrzeć, należy nacisnąć FIRE i wychylić joystick w lewo lub prawo. Na każdej stronie Księgi znajduje się nazwa zaklęcia i nazwy ziół lub przedmiotów potrzebnych do przygotowania mikstury pozwalającej używać danego czaru. Obok Księgi znajdują się dwa sarkofagi symbolizujące zapas energii obu braci. Prawy dolny róg to „kompas” wskazujący część wioski, w której przebywa nasz wróg — Leanoric.

Gdy już będziesz miał dwa zioła potrzebne do jednego zaklęcia, musisz odnaleźć swój czarodziejski kotł. (Claudron — zwróć na niego uwagę przy rozpoczęciu gry). Podejdź do kotła i otwórz Księgę na odpowiedniej stronie. Zostanie przyrządzona mikstura, a ty będziesz mógł używać tego czaru. Aby tego dokonać, musisz otworzyć Księgę na stronie z danym zaklęciem i nacisnąć FIRE. Pamiętaj jednak, że twój przeciwnik również zbiera zioła, znosi je do swojego kotła i przygotowuje zaklęcia — musisz się spieszyć.

Księga Czarów ma 12 stron; na każdej stronie znajduje się nazwa zaklęcia i nazwa dwóch ziół potrzebnych do wejścia w posiadanie czaru. Oto lista zaklęć z zakresem ich działania:

TELEPORT — pozwala na przeniesienie Learica do jego domu (a tym samym w pobliże czarodziejskiego kotła). Zaklęcia można używać dowolną ilość razy.

LIGHTNING — pozwala na trzykrotne wypuszczenie błyskawicy, która sama trafi wroga, jeżeli ten jest na ekranie.

FIREBALL — działa podobnie jak błyskawica, tylko że rzucasz ognistymi kulami.

HEAL — pozwala na zwiększenie zasobów twojej energii. Działa dwukrotnie.

REVERSE — pozwala na odwrócenie ruchów postaci. Działa trzykrotnie.

INVISIBLE — pozwala na pewien czas stać się niewidzialnym. Działa trzykrotnie.

DOPPLEGANGER — pozwala na stworzenie twojego sobowtóra. Działa trzykrotnie.

FREEZE — unieruchamia na pewien czas przeciwnika. Działa trzykrotnie.

SWIFT — przyspiesza ruchy Learica. Działa trzykrotnie.

ZOMBIE — pozwala na zamienienie człowieka w kapturze i z widłami na lunatyka zależnego od naszych ruchów i nękającego wroga. Działa trzykrotnie.

SPRITES — pozwala na utworzenie duszków, kilku czasek, które przez pewien czas nękają wroga. Działa tylko jeden raz.

PROTECT — pozwala na pewien czas uchronić Learica przed czarami wroga. Działa trzy razy.

(wist)

Ocena (w skali od 1 do 10):

GRAFIKA 8

DŹWIĘK 7

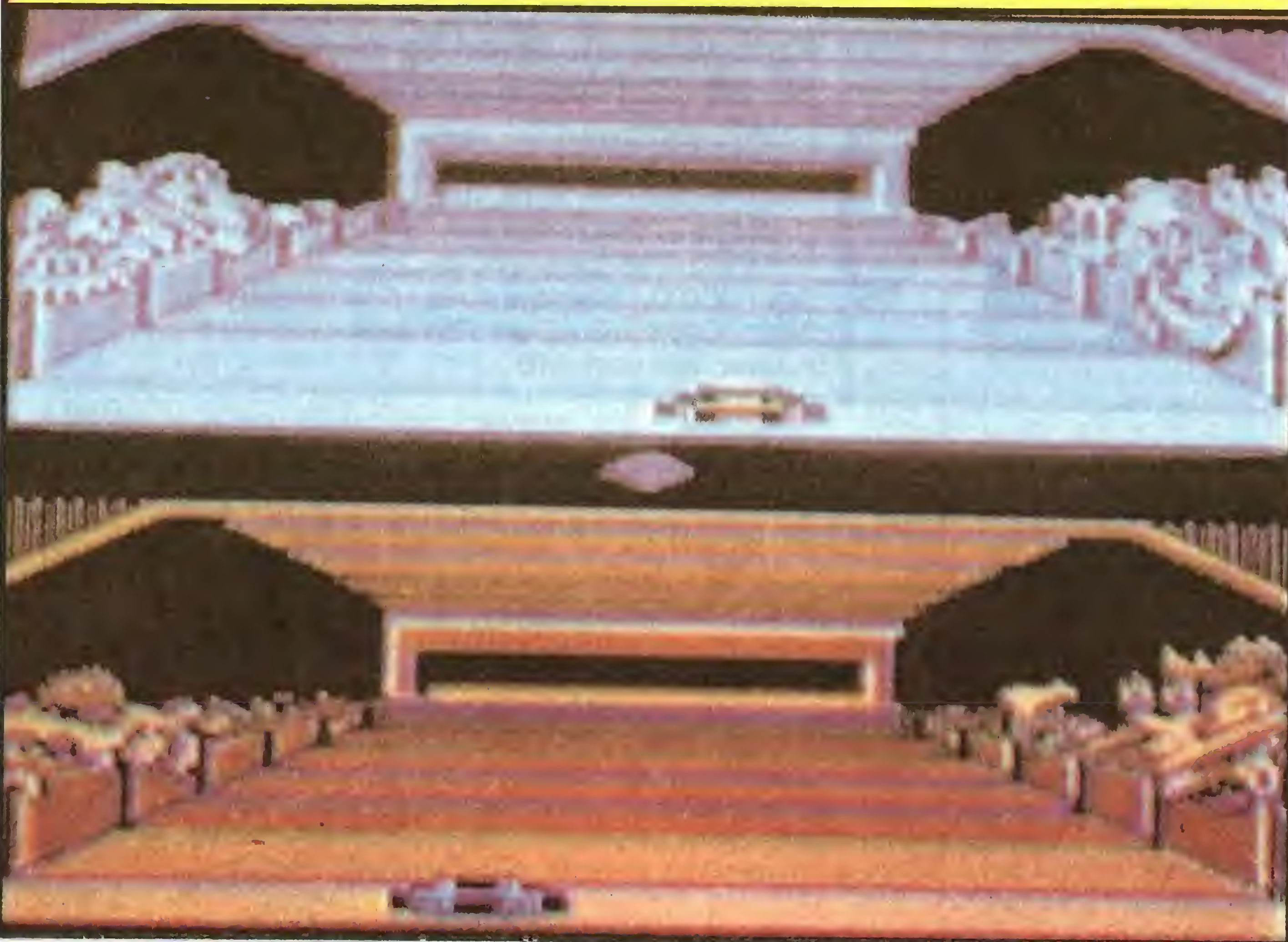
SENS GRY 8

Producent: Bulldog Software

Autor: Trevor Brown

REBOUND

XL/XE



Bez wątpienia

rynek gier komputerowych od pewnego czasu cierpi na brak oryginalności. Co chwila okazuje się, że programiści po prostu zmieniają grafikę i sprzedają gry typu „zabij-uciekaj” pod zmienionym tytułem. Kto wie, może i tym razem mamy do czynienia ze starym, odświe-

żonym pomysłem, ale popatrzmy nań nieco dokładniej.

Świetnie zrobiony tytułowy obrazek, na którym nie ma żadnych zabawnych efektów, ale za to tekst napisany wieloma krojami pisma w połączeniu z kolorami daje dobry efekt.

Muzyka, a szczególnie symulacja perkusji, powoduje, że trudno oprzeć się chęci wystukiwania rytmu na stole. Zachwyciło mnie, że nie jest to monotonna fraza muzyczna, ale trwający kilka minut (sic!) utwór. W czasie samej gry również słychać tę muzykę w tle — osobiście bardzo to lubię.

Teraz przejdźmy do samej gry. Wyobraź sobie kort tenisowy, przecięty na pół wzdłuż siatki, a jego połówki w górnej i dolnej części ekranu.

Na każdym polu znajduje się rakietę gracza, którą można przesuwac w płaszczyźnie poziomej. Boki kortu obudowane są bandami, aby piłka mogła się od nich odbijać. Ot, prawie wszystko — wygląda, że to banalne?

Dostępne są trzy poziomy trudności — pierwszy wręcz wymarzony dla początkujących, ale nie radzę próbować sił na trzecim, jeśli nie masz refleksu Borisa Beckera.

Wybierając opcję „one player” zmierzysz się z komputerem, ale prawdziwa zabawa zaczyna się dopiero gdy „zatrudnisz” jako przeciwnika swego kolegę.

Celem gry jest zdobycie siedmiu punktów (a może goli?). Zdobywasz je wtedy, gdy przeciwnik nie odbierze piłki, ale pod warunkiem, że akurat byłś stroną serwującą — jeśli nie, to przeciwnik tylko traci serw. Wynik pokazywany jest na linii dzielącej obie połówki kortu. Gdy gra się kończy, rozlega się gwizdek i powracasz do tytułowego ekranu.

Niezwykle zabawna jest animowana publiczność, przyglądająca się rozgrywce z obu stron kortu. Składa się na nią najdziwniejsza zbieranina przeróżnych stworów.

Gdy tylko gracz, którego są kibicami, zdobywa punkt lub serw, tak bardzo się cieszą, że podskakują, klaszczą, wrzeszczą, zwijają się z wielkiej radości.

Gra ma wszystkie cechy, by stać się przebojem — płynne przewijanie, doskonałą grafikę i wspaniałą muzykę. To wszystko powoduje, że gra jest niezwykle wciągająca i chętnie powraca się do swojej animowanej publiczności.

(wist)

Ocena (w skali od 1 do 10):

GRAFIKA 8

DŹWIĘK 10

SENS GRY 7

Producent MicroValue/Tyresoft

Autor: Ivan MacIntosh

BAKER STREET 221-B

XL/XE



Podtytuł tej gry powinien brzmieć: szukaj wskazówek, znajdź mordercę, określ broń i wydedykuj motyw. W samej grze dzieje się znacznie więcej.

Zabawa oparta jest na grze planszowej o tym samym tytule co nasz program. Miejsce akcji jest zamglony Londyn. W mieście tym zdarza się bardzo wiele zbrodni, których tajemnice może odkryć jedynie Sherlock Holmes lub ktoś z jego bliskich współpracowników.

W grze może brać udział od 1 do 4 osób. Każda wciela się w jedną z następujących postaci: Sherlock Holmes, Doktor Watson, Inspektor Lestrade, a nawet Irene Adler.

Możemy sobie utrudnić życie godząc się na to, aby wszelkie wskazówki, jakie znajdujemy w grze, były zakodowane. Pozostaje wybrać jeden z trzydziestu dostępnych scenariuszy zbrodni i do dzieła. Dodam jeszcze, że warto wyposażać się w kartkę, na której będziemy zapisywać wszystkie zdobyte informacje. Przydatny może się okazać również (niestety) słownik angielsko-polski.

Po naciśnięciu klawisza „M” ukaże się mapa miasta ze wszystkimi budynkami i z zaznaczeniem miejsca, gdzie znajdują się poszczególni gracze. Na początku wszyscy znajdują się na Baker Street 221-B. Tu też wędrówka się

kończy. Tymczasem kolejno każdy, naciskając „fire” lub klawisz spacji, symuluje rzut kostką i porusza się o odpowiednią liczbę kroków w dowolnym możliwym kierunku. Do poruszania się można używać zarówno joysticka, jak i klawiatury. W przypadku tej ostatniej stosuje się klawisze „I” — ruch do góry, „K” — ruch do dołu, „J” — w lewo, „L” — w prawo. Wejścia do budynków oznaczone są dużymi strzałkami. W każdym z nich znajdziemy jakąś wskazówkę, informację lub pomoc. Konieczna jest odznaka (badge), którą możemy dostać w budynku Scotland Yardu. Bez niej nie można przystąpić do końcowego quizu. Pozwala ona również na „zamykanie” dowolnych budynków, tak aby inni uczestnicy gry nie mieli dostępu do informacji w nich zawartych. Ważny jest także klucz (do zdobycia u ślusarza — „locksmith”). Przydaje się do otwierania zamkniętych przez innych budynków i wchodzenia do podziemnego tunelu, który szybko przenosi gracza do losowo wybranego miejsca w mieście. Warto pamiętać, że użycie każdego z tych atrybutów możliwe jest tylko raz, po czym trzeba zdobyć je ponownie.

Bardzo użyteczny jest „Carriage Dept.” — można tu wynająć dorożkę i szybko dostać się nią do każdego punktu miasta.

Gdy zdobędziesz już odznakę oraz dość informacji (im mniej tym lepiej), aby dokładnie określić zbrodnię, udaj się z powrotem na Baker Street i przystąp do końcowego quizu. Jeśli w którejś z części pomylisz się, znów znajdziesz się na ulicy, już bez odznaki, a program powie ci, ile miałeś błędnych odpowiedzi.

Program wyposażono w niezłe efekty dźwiękowe — nawet syntezę mowy i onomatopeje. Grafika jest również dobra, choć trudno powiedzieć o niej, by zapierała dech. Myślę, że zagadki mogą okazać się niezłym ćwiczeniem dla umysłu, choć dla prawdziwych smakoszy fabuł kryminalnych mogą okazać się banalne.

(wist)

Ocena (w skali od 1 do 10):

GRAFIKA 7

DŹWIĘK 6

SENS GRY 8,5

PRODUCENT: Datasoft

Autor: Steven Duboff

Oficjalny dystrybutor oryginalnego, polskiego oprogramowania do komputerów

SPECTRUM, TIMEX,
ATARI,
COMMODORE,
IBM

oferuje gry i programy użytkowe sklepom, klubom, studiom komputerowym i wypożyczalniom.

Poszukujemy osób zainteresowanych współpracą w rozpowszechnianiu oprogramowania.

Przyjmujemy zlecenia na pisanie programów.

Nasze programy reprezentują profesjonalny poziom wykonania oraz posiadają ochronę prawną. Najlepsze z nich mają zapewnioną reklamę w szerokim kręgu odbiorców.

SPECTRA
21-422 Stanin
tel. 11-70

B 65

ATARI 800 XL, 65 XE, 130 XE

Sprzedaż wysyłkowa
gier i programów użytkowych
na kasetach i dyskietkach.
Również w systemie TURBO
2000
Wszystkie nowości!!!
Instrukcje i literatura.
Dla zainteresowanych rachunki.

ANWIKOL

03-721 Warszawa
ul. Jagiellońska 3/23.

SB74

SERWIS KOMPUTERÓW

TEST

**Katowice, ul. Armii Czerwonej 22/53 tel.
59-83-22 (superjednostka) IX piętro**

poleca naprawy:

Atari 600, 800, 65, 130, XL, XE
Commodore 16, 116, +4, 64, 128, 1280, Amiga
Disk Drive 1541, 1570, 1571, 1050

rozszerzenia pamięci:

Atari 600XL, Commodore 16, 116 do 64 kB

Atari 800XL, 65XE do 130 kB

Amiga 500 do 1 MB

godz. 9—11, 15—18

SB30

• ATARI •

ATARI 800 XL
65 XE, 130 XE

Literatura, opisy
oprogramowania,
katalogi gratis.

Sprzedaż wysyłkowa

TURBO 2000

Warszawa 04-357
Grzechowska 186 m 69

A2

MÓZG PROCESOR!

to rewelacyjna
polska gra przygodowa firmy

COMPUTER
ADVENTURE STUDIO

— dla **SPECTRUM, TIMEX-a,**
JUNIORA (taśma + opis)
— dla **ATARI XL/XE** (dyskietka
+ opis)

Cena zestawu
— równoważność 2 USD
UWAGA! Test w Bajtku nr 10/89
Zamówienia prosimy kierować:
COMPUTER ADVENTURE STUDIO
Bocnia 32-700,
Kazimierza Wielkiego 37/45
tel. (0-187) 242-47

B4

Reklamuj się w „MOIM Atari”

Nasz magazyn dociera do wszystkich użytkowników Atari. Jeśli i Ty chcesz tam dotrzeć, reklamuj się w „Moim Atari”.

Cena 1 cm² - 10 tys. zł, cała strona - 10 mln zł.

Ogłoszenie kolorowe - 100% drożej.

Minimalny format ogłoszenia - 20 cm².

Ogłoszenia przyjmuje Biuro Reklamy Spółdzielni „Bajtek”, Warszawa, ul. Wspólna 61, tel. 21-12-05.

**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



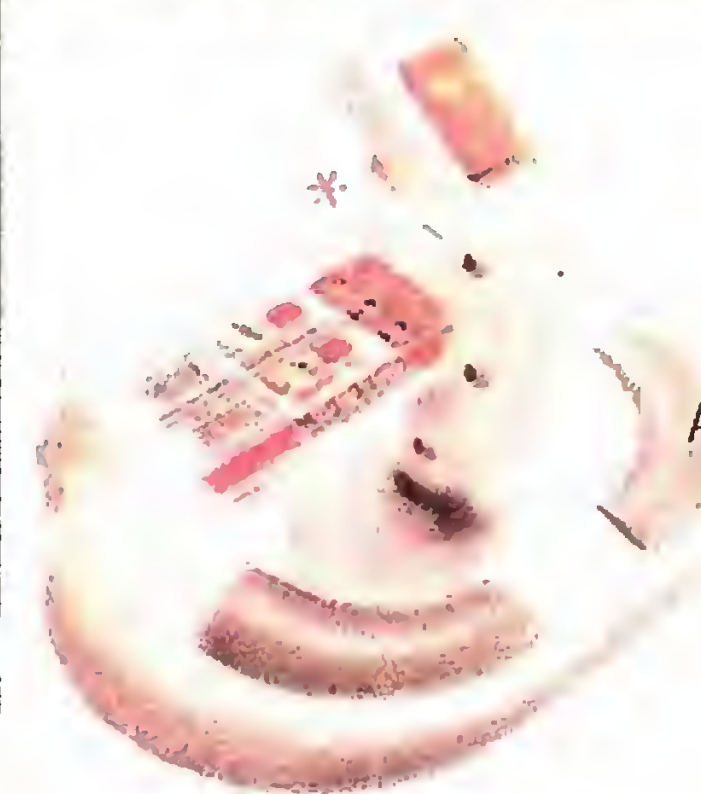
**SV 125
Superboard**

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 T/O Card
2 Fire
2 AutoFire
6 Mikrostryków
Wybór AUTO
PSC — Regulator XY
Sygnalizacja Światła
Fire
ASC — Regulator szybkości
AUTO



**SV 202
M 6 analog**

Analogowy
DO IBM XT/AT
(kompatybilnych)
Współpracuje z Game-Card
lub T/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
Platinowane 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 140 — 389 000
SV 201 — 249 000

SV 202 — 229 000
SV 210 — 219 000
SV 500 — 119 000
SV 510 — 119 000

**sprzedaż hurtowa
tel. 23-98-53**

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

**udziela informacji
tel. 662-35-16**

JAK ZAMAWIAĆ? Należy przelać pieniądze przekazem telegraficznym (duży zielony blankiet) wraz z dokładnym symbolem joysticka na adres:

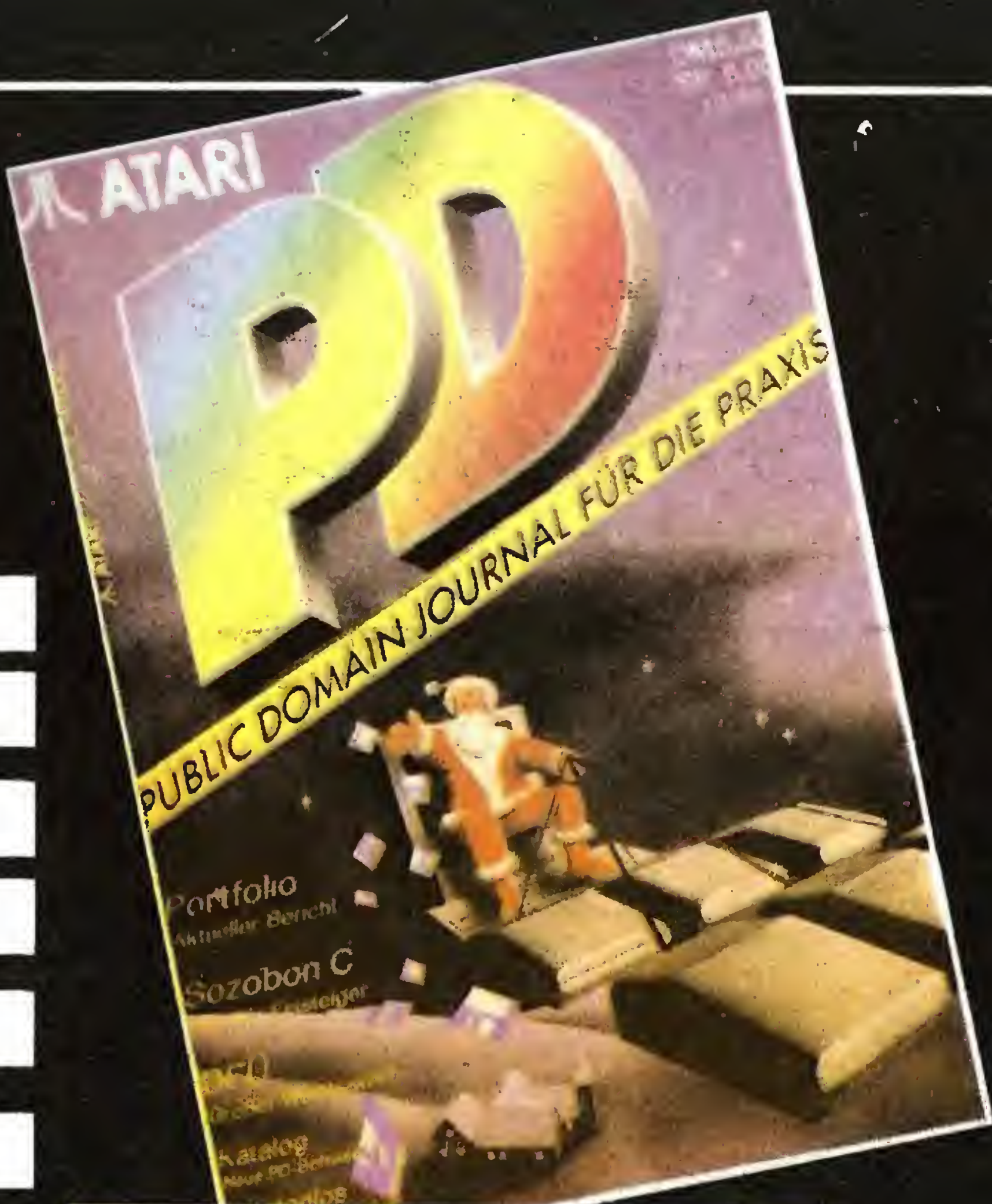
TAL

„Tal” Sp. z o.o.
ul. Mikowa 45
02-411 Warszawa 21
fax: 659-12-35

Quickjoy

„TAL” Sp. z o.o.
Skr. poczt. 51
02-105 Warszawa 21

Czas realizacji 3—6 dni



„ATARI MAGAZIN”

To pismo nie jest tak popularne, jak „ST Computer”, czy „ST Magazin”. Ma jednak swoje zalety. „Atari Magazin”, w odróżnieniu od swoich wielkich konkurentów, prowadzi spory kącik informacyjny dla użytkowników Atari XL/XE. Okazuje się, że nawet w RFN jest niemała grupa sympatyków 8-bitowego Atari.

Miesięcznik ten jest adresowany do użytkowników 8- i 16-bitowych komputerów. Znajdziemy w nim przegląd nowości oprogramowania, informacje techniczne, listy programów, ciekawostki, krótkie opisy gier, wywiady z ciekawymi ludźmi. Prowadzone są tu także kursy programowania w różnych językach. Redakcja udostępnia również oprogramowanie „public domain” dla ST i XL/XE. Pismo stara się utrzymywać ożywione kontakty z czytelnikami. Giełda ogłoszeń służy jako skrzynka kontaktowa.

O skromniejszej randze „Atari Magazin” świadczy stosunkowo niewielka liczba reklam producentów sprzętu i oprogramowania. Mimo to pismo jest wydawane starannie i fachowo, co z pewnością zjednało mu licznych sympatyków.

„Atari Magazin”, 100 stron, cena 7 DM.
Verlag Raetz, Postfach 1640, 7518 Bretten, BRD

„ANTIC”

Ten miesięcznik jest wydawany w Stanach Zjednoczonych już od ponad ośmiu lat. Jest on przeznaczony, o czym świadczy nazwa, wyłącznie dla użytkowników 8-bitowego Atari.

Styl pisma zdecydowanie różni się od jego niemieckich odpowiedników. Nie ma w nim wszedobylskich reklam. Publikowane są naprawdę nieliczne ogłoszenia, oczywiście tylko te, które oferują sprzęt i oprogramowanie do Atari XL/XE.

Podstawowym założeniem „Antica” jest publikowanie wszelkich wiadomo-

Prezentowane przez nas czasopisma komputerowe stanowią tylko niewielką część tytułów ukazujących się na Zachodzie. Każde z nich ma swój odrębny styl, dzięki czemu czytelnik może wybrać to, co mu najbardziej odpowiada.

ści dotyczących oprogramowania 8-bitowego Atari. Jest on znakomitą pomocą dla osób specjalizujących się w programowaniu, każdy z artykułów zawiera bowiem bardzo starannie i wyczerpująco opracowane informacje źródłowe. Około 1/3, a czasami nawet połowę każdego numeru zajmują listy omawianych tu programów. Jest to świetna nauka dla wszystkich, którzy pragną biegle poznać technikę programowania. Oczywiście, jeśli nuży nas przepisywanie tekstu listingu na własnym komputerze, możemy zamówić dyskietkę z gotowym programem.

A co zrobić, jeśli zamienimy 8-bitowe Atari XL lub XE na 16-bitowe ST? Redakcja ma coś w zanadrzu i dla tych czytelników. Następca „Antica” jest miesięcznik „STart”, który wprowadzi nas w świat ST.

„Antic”, 48 stron, cena 3,95 \$.
544 Second Street, San Francisco, CA 94107, USA

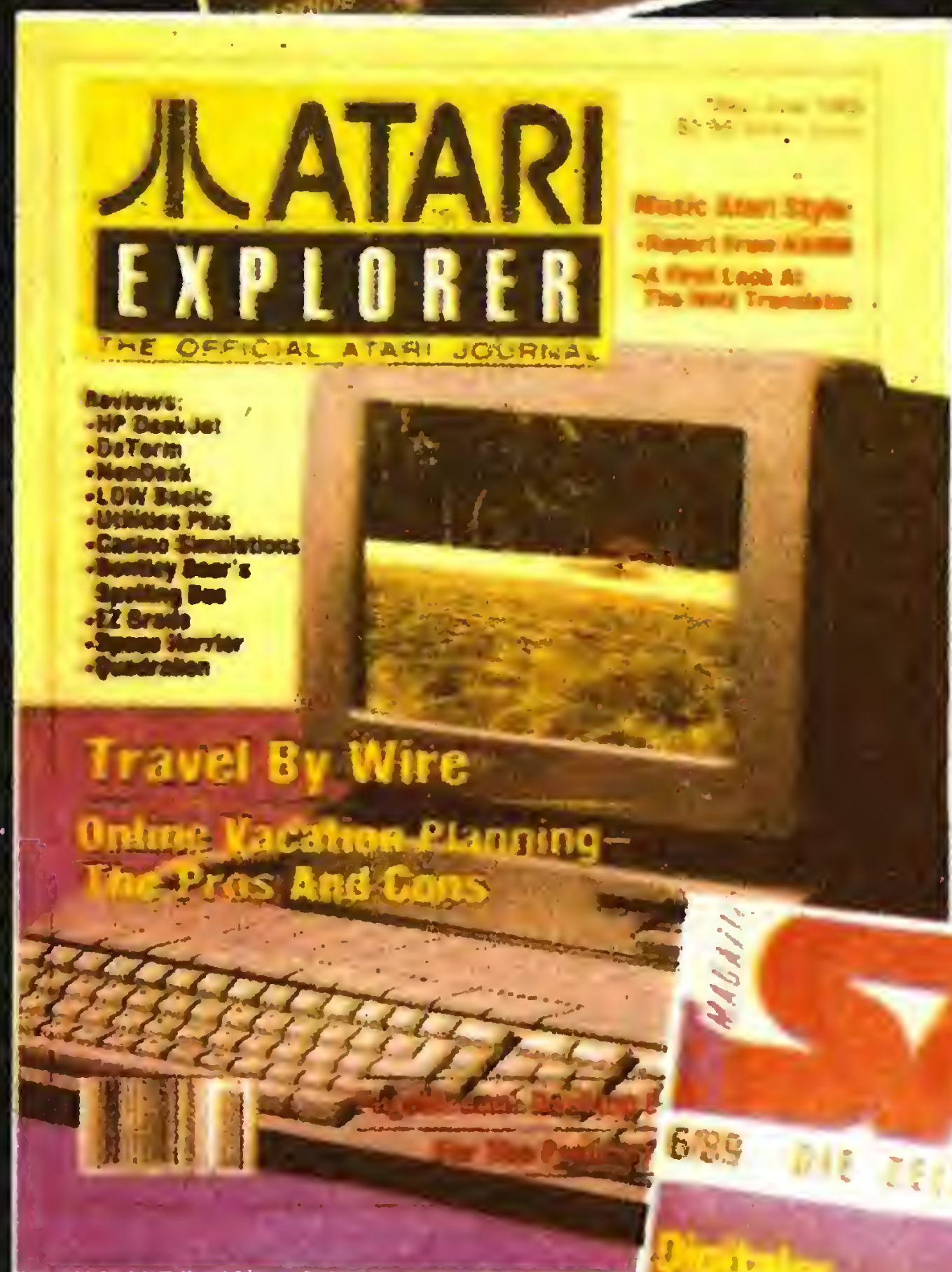
„NEW ATARI USER”

„New Atari User” wydawany jest w Anglii, a powstał z połączenia wcześniejszych pism „Atari User” i „Page 6”. Znajdziemy w nim mieszankę wszystkiego, co może zainteresować użytkowników Atari XL/XE oraz ST, przy czym większa część numeru jest zwykle poświęcana komputerom 8-bitowym.

Pismo jest przeznaczone raczej dla początkującego użytkownika. Dominują w nim opisy gier. Nie brakuje tu także listingów różnych programów i procedur, które napisane są przeważnie w Basicu, assemblerze lub Pascalu.

Spora rubryka jest poświęcona edukacji. Prezentowane są tu przede wszystkim programy. Taka metoda nauki pozwala przy świetnej zabawie poznawać jednocześnie geografię, historię lub inne przedmioty.

W całym numerze znajduje się bardzo niewiele ogłoszeń. Pismo prowadzi skrzynkę kontaktową dla czytelników, którzy chcą sprzedać sprzęt i programy lub poszukują czegoś do swojego komputera.



ST MAGAZIN

„ST Magazin” jest — obok „ST Computer” — podstawową lekturą użytkowników ST w RFN. W piśmie tym znajdziemy wyczerpujące informacje o nowościach z dziedziny sprzętu i oprogramowania. Redakcyjni fachowcy dzielą się swoimi uwagami na temat zalet i wad programów oraz prezentowanych urządzeń.

Numer „ST Magazin” zwykle zawiera jakiś temat wiodący, np. grafika, monitory, przetwarzanie tekstów. Dzięki temu każde wydanie staje się swego rodzaju przewodnikiem po danym zagadnieniu.

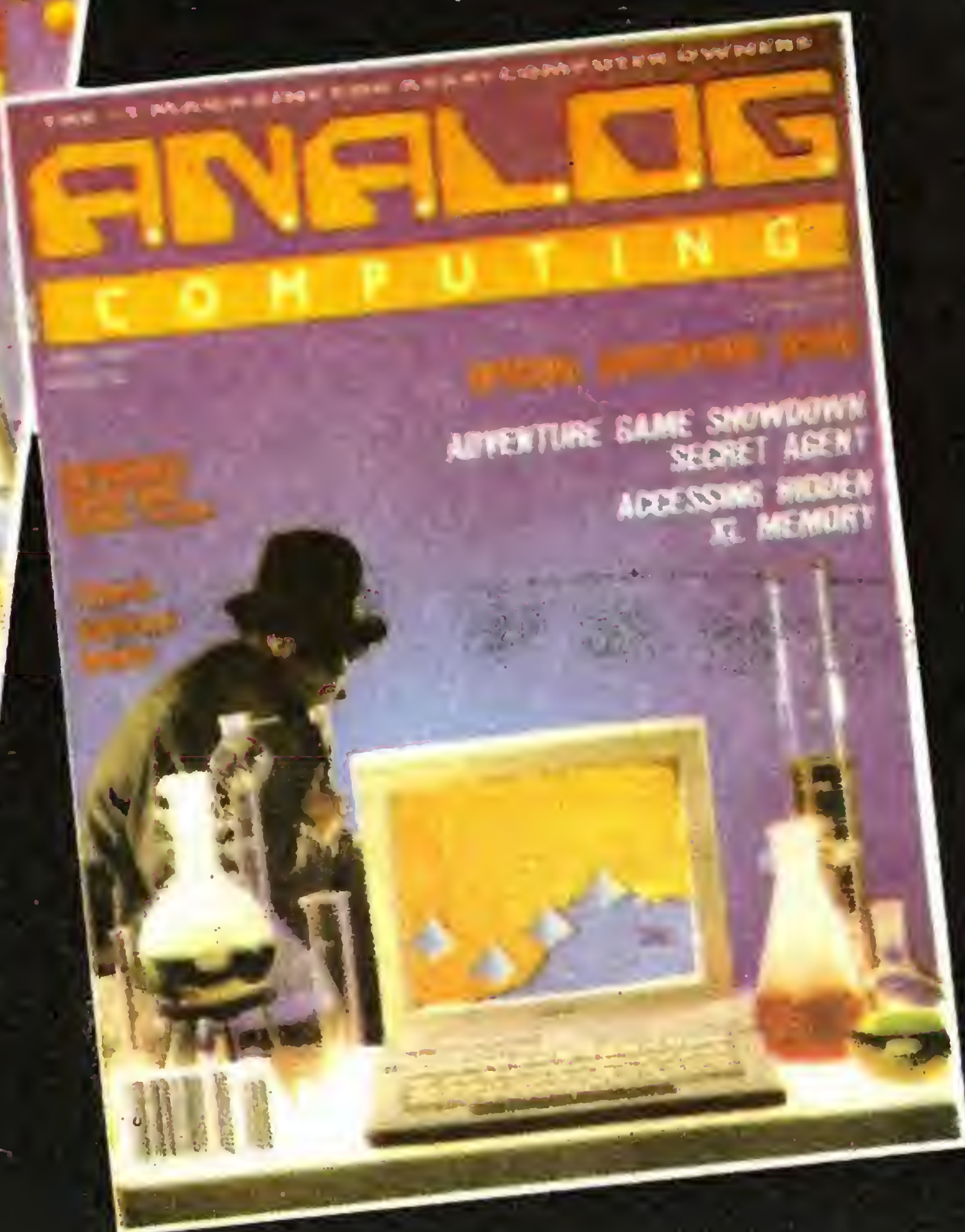
W redakcyjnym kąciku pierwszej pomocy czytelnik może uzyskać odpowiedź na nurtujące go pytania.

Na środkowej stronie niektórych numerów znajdują się zwięzłe opisy — ściągawki ułatwiające posługiwanie się językami programowania i programami użytkowymi. Z tych wkładek można skompletować poradnik niezwykle pożyteczny dla każdego programisty.

Podobnie jak w większości niemieckich czasopism, na kolumnach „ST Magazin” królują ogłoszenia. Liczne firmy oferują dosłownie wszystko, co jest potrzebne użytkownikowi komputera: sprzęt, programy, literaturę fachową, dyskietki, pudełka itp.



ST Magazin, 160 stron, cena 7 DM.
ST Magazin Leser-Service, Markt & Technik Verlag, Hans-Pinselstr. 2, 8013 Haar bei Muenchen, BRD



Lista tytułów zmienia się bardzo szybko. Corocznie powstają nowe typy komputerów, które zdobywają zwolenników. Redakcje muszą nadążyć za postępem technicznym, by zdobyć jak najwięcej czytelników.

„New Atari User”, 76 stron, cena 1,50 \$.
P.O. Box 54, Stafford, ST16 1DR,
England, tel. 0785 213928

„ST COMPUTER”

„ST Computer” jest jednym z najpopularniejszych niemieckojęzycznych pism dla użytkowników ST. Z tego magazynu można dowiedzieć się wszystkiego, co dzieje się na rynku oprogramowania i sprzętu. W każdym numerze najwięcej miejsca przeznaczają się na zaprezentowanie czytelnikom wszelkich nowości z dziedziny oprogramowania. Możliwości każdego programu użytkowego są wnikliwie testowane przez redakcję, by czytelnik mógł rzetelnie ocenić jego przydatność do swojego warsztatu pracy. „ST Computer” jest też bardzo atrakcyjny dla programistów poszukujących pomysłów lub gotowych rozwiązań pewnych problemów. Przewidziano dla nich duży kącik zawierający listingi najrozsądniejszych procedur użytkowych i krótkich programów.

W poważnym magazynie nie może oczywiście zabraknąć informacji o sprzęcie. Są tu testy, opisy drukarek, modemów, stacji dysków i oczywiście samych komputerów.

W „ST Computer” reklamują się wszystkie firmy, które mają cokolwiek do powiedzenia na rynku ST.

„ST Computer”, ok. 196 stron, cena 7 DM
Heim Verlag, Heidelberger Landstr.
194, 6100, Darmstadt 13, BRD

„ATARI PD JOURNAL”

Użytkownik Atari ST ma do dyspozycji ogromną bibliotekę oprogramowania „public domain”, to znaczy procedur i programów, którymi każdy może bezpłatnie się posługiwać. O tym, jak dużo jest tych programów i jak ważną rolę spełniają w codziennej pracy z komputerem, świadczy fakt, że wydawca „ST

Computer” zdecydował się na publikowanie osobnego pisma poświęconego tej tematyce.

„Atari PD Journal” jest swego rodzaju katalogiem oprogramowania „public domain”. Czytelnik znajdzie tu opisy gier, programów narzędziowych, edukacyjnych, muzycznych i wielu innych. Redakcja dysponuje, w sumie, ponad dwoma tysiącami programów na przeszło trzystu dyskietkach. Każdy z nich można zamówić po cenie odpowiadającej praktycznie kosztom dyskietki i wysyłki do klienta.

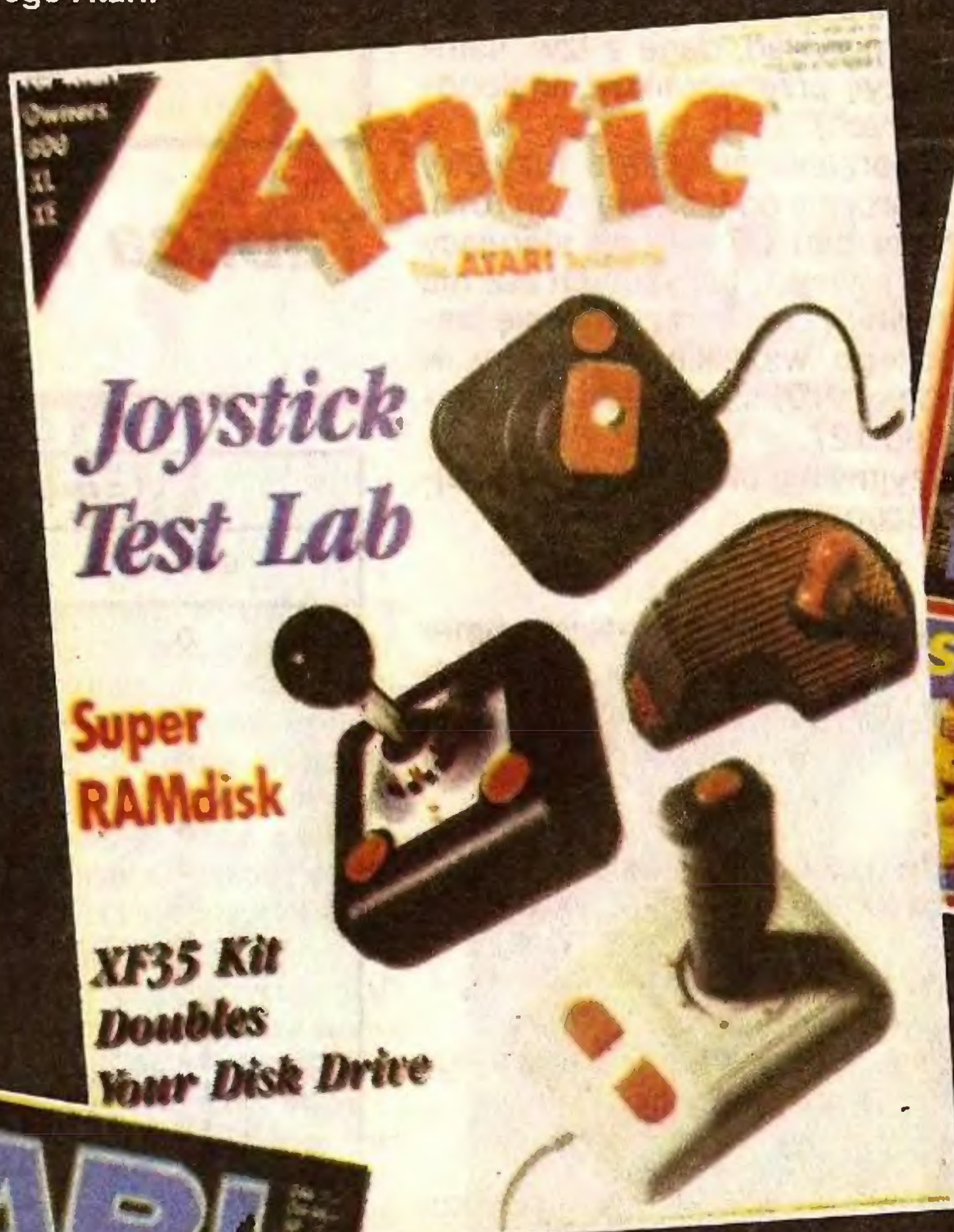
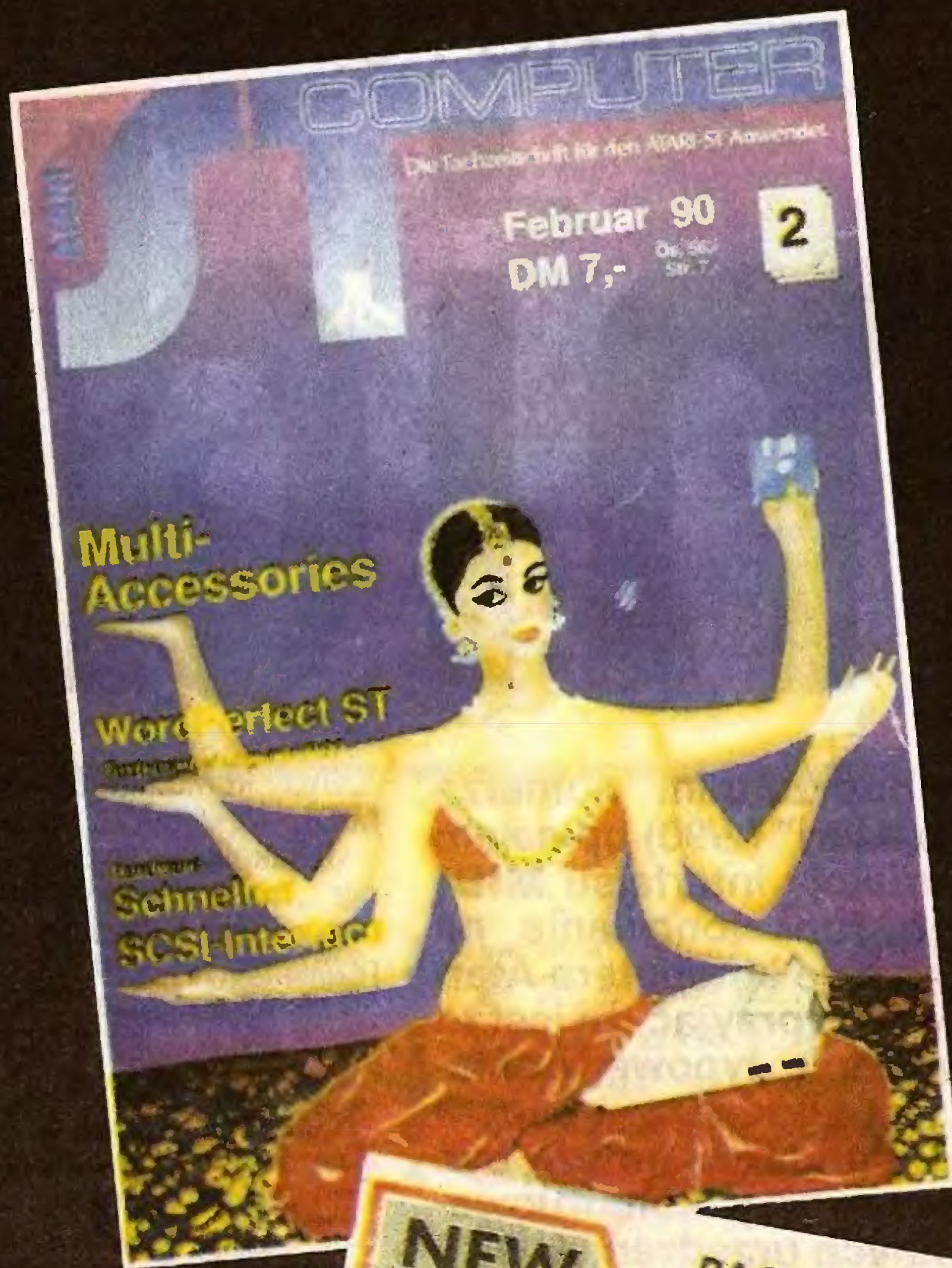
Podstawowym celem, jaki stawia sobie „Atari PD Journal”, jest oczywiście pomóc czytelnikowi dotrzeć do najcenniejszych programów z tej ogromnej biblioteki „public domain”. Użytkownicy często nie zdają sobie sprawy z tego, jak wiele korzyści mogą odnieść z dyskietek, które uzyskują praktycznie w prezencie.

Atrakcyjność pisma podnoszą dodatkowo napisane w bardzo przystępny sposób opisy sprzętu, nowinek technicznych, konkursy dla czytelników oraz giełda bezpłatnych ogłoszeń drobnych.

„Atari PD Journal”, 76 stron, cena 5 DM
Heim Verlag, Heidelberger Landstr.
194, 6100 Darmstadt 13, BRD

„ATARI EXPLORER”

„Atari Explorer” jest wydawany w Stanach Zjednoczonych i w Kanadzie. Czasopismo to jest adresowane przede wszystkim do użytkowników ST, lecz swój niewielki kącik mają również i sympatycy 8-bitowego Atari.



„ANALOG COMPUTING”

„Analog Computing” reklamuje się następującym hasłem: „Masz komputer z mikroprocesorem 6502. Po co mamy cię zmuszać do czytania magazynu, którego połowa jest poświęcona Atari ST”.

„Analog Computing” jest skierowany do użytkowników Atari XL/XE, a szczególnie do tych, którzy lubią programować i chcą się tego uczyć. Pismo zawiera prawie wyłącznie listingi programów i cykle artykułów uczących programowania w różnych językach bądź posługiwania się systemem operacyjnym lub profesjonalnymi programami narzędziowymi. Opisane są tu nie tylko programy „public domain”. Niektóre z nich to programy znajdujące się w handlu, a ich teksty źródłowe udostępniono wyłącznie czytelnikom „Analog Computing”. Oczywiście wraz z pismem można zamówić również dyskietkę.

Część programów publikowanych w „Analog Computing” została napisana przez jego czytelników. Jeśli ma się ciekawy pomysł, w każdej chwili można zostać współautorem czasopisma.

Prenumerata „Analog Computing” z dyskietką jest znakomitym sposobem na systematyczne wzbogacanie biblioteki oprogramowania.

Pismo ma wyraźnie określony profil tematyczny. Jego podstawowym celem jest zaprezentowanie czytelnikowi oprogramowania, zachęcenie go do posługiwania się komputerem i przekonanie o korzyściach płynących z tej decyzji.

Trzeba przyznać, że redakcja realizuje to założenie naprawdę dobrze. Na przykład w kilku kolejnych numerach w 1989 roku ukazywał się cykl artykułów poświęconych sieci CompuServe. Jej abonenci mogą, nie wychodząc z domu, załatwić prawie wszystkie sprawy: zaplanować podróż, zarezerwować hotel i bilety na samolot, zrobić zakupy, nawiązać kontakt z przyjaciółmi w odległym zakątku kraju.

„Atari Explorer” pisze nie tylko o sieci CompuServe. Znajdziemy tu opisy różnych programów użytkowych, muzycznych, kalkulacyjnych, desktop publishing. W piśmie prezentowane i oceniane są również gry. Kącik programowania jest bardzo skromny, nie więcej niż 2-3 listingi. Bardzo duży wpływ na treść pisma ma fakt, że jest to oficjalne wydawnictwo firmy Atari Corp.

„Atari Explorer”, 82 strony, cena 2,95\$.
Atari Explorer, CN961, Netcong, NJ
07857-0961, USA

„Analog Computing”, 84 strony, cena 3.95\$. P.O. Box 16927, N. Hollywood, CA 91615-9966, USA

Janusz Jarmoch

PROTOKÓŁ

midi

W poprzednim numerze „Moje Atari” (2/90) opisałem konstrukcję interfejsu MIDI i jego funkcjonowanie na przykładzie komputera Atari ST. Można przyjąć, że jest to rozwiązanie typowe, ponieważ jest zgodne ze specyfikacją standardu i może być używane przy konstruowaniu innych urządzeń MIDI.

Teraz chciałbym opisać protokół MIDI, czyli język którego używają połączone tym interfejsem urządzenia. Opis będzie na tyle szczegółowy, aby umożliwić wszystkim zainteresowanym pisanie własnych programów. Czekamy na efekty Waszej działalności!

Protokół został zredagowany w taki sposób, aby uniknąć przekazywania łączem MIDI jakiegokolwiek zbędnej informacji. W tym celu, na przykład, przypisano każdemu dźwiękowi skali muzycznej w obrębie dziesięciu oktaw odpowiednią liczbę (kod MIDI) z przedziału od 0 do 127. Oznacza to, że maksymalna klawiatura obsługiwana przez MIDI może zawierać 128 klawiszy (dla porównania — klawiatura pianina zawiera do 88 klawiszy). Tablica 1 przedstawia wszystkie kody MIDI określające wysokość dźwięku.

W skali od 0 do 127 określono również głośność generowanych dźwięków; liczba 0 oznacza brak dźwięku a 127 — maksymalne jego natężenie.

W opisie standardu przypisano także kody cyfrowe (od 0 do 127) wszystkim typowym przełącznikom i regulatorom znajdującym się w większości profesjonalnych syntezatorów, w tym również przełącznikom banków brzmień, których może być maksymalnie do 127. Tablica 2 określa kody MIDI wszystkich przewidzianych w standardzie regulatorów wpływających na brzmienie generowanych dźwięków.

Podane powyżej informacje nie wyczerpują, oczywiście, całości tematu, ale chwilowo poprzestaniemy na tym, aby nie zaciemniać obrazu. Skoncentrujemy się najpierw na sposobie tworzenia komunikatów w standardzie MIDI.

Jakakolwiek zmiana stanu urządzenia „master”, czyli nadającego (np. naciśnięcie klawisza, uruchomienie przełącznika), jest sygnalizowana wysłaniem komunikatu, który na oscyloskopie wyglądałby mniej więcej tak, jak przedstawia to rysunek 1.

Oczywiście, bity START i STOP są ignorowane przy interpretacji komunikatu. Pierwszy bajt to bajt sterujący (COMMAND BYTE), po nim zaś następują jeden lub dwa bajty danych (DATA BYTE). Jest to zasada, od której istnieją dwa wyjątki: mogą być nadawane komunikaty jednobajtowe (np. MIDI CLOCK) lub mogą być nadawane dowolnie długie ciągi danych w trybie komunikatu SYSTEM EXCLUSIVE (są to, na przykład, dane z tzw. samplów, czyli przetworników analogowo-cyfrowych).

Przy tworzeniu standardu przyjęto, że bajt sterujący od bajtu danych różni się stanem bitu D7 — bajt sterujący ma bit D7 równy 1, bajt danych zaś ma bit D7 równy zero. Teraz staje się jasne, dlaczego wszelkie parametry w standardzie MIDI są określane liczbami od 0 do 127.

Przyjrzyjmy się bliżej strukturze bajtu sterującego:

D0	te cztery bity określają numer kanału, dla którego przeznaczony jest dany komunikat;
D1	
D2	
D3	
D4	te trzy definiują osiem podstawowych komunikatów MIDI;
D5	
D6	
D7	identyfikator bajtu sterującego (zawsze = 1).

a oto owe osiem podstawowych komunikatów:

D6	D5	D4	
0	0	0	— NOTE OFF
0	0	1	— NOTE ON
0	1	0	— POLYPHONIC KEY PRESSURE
0	1	1	— CONTROL CHANGE
1	0	0	— PROGRAM CHANGE

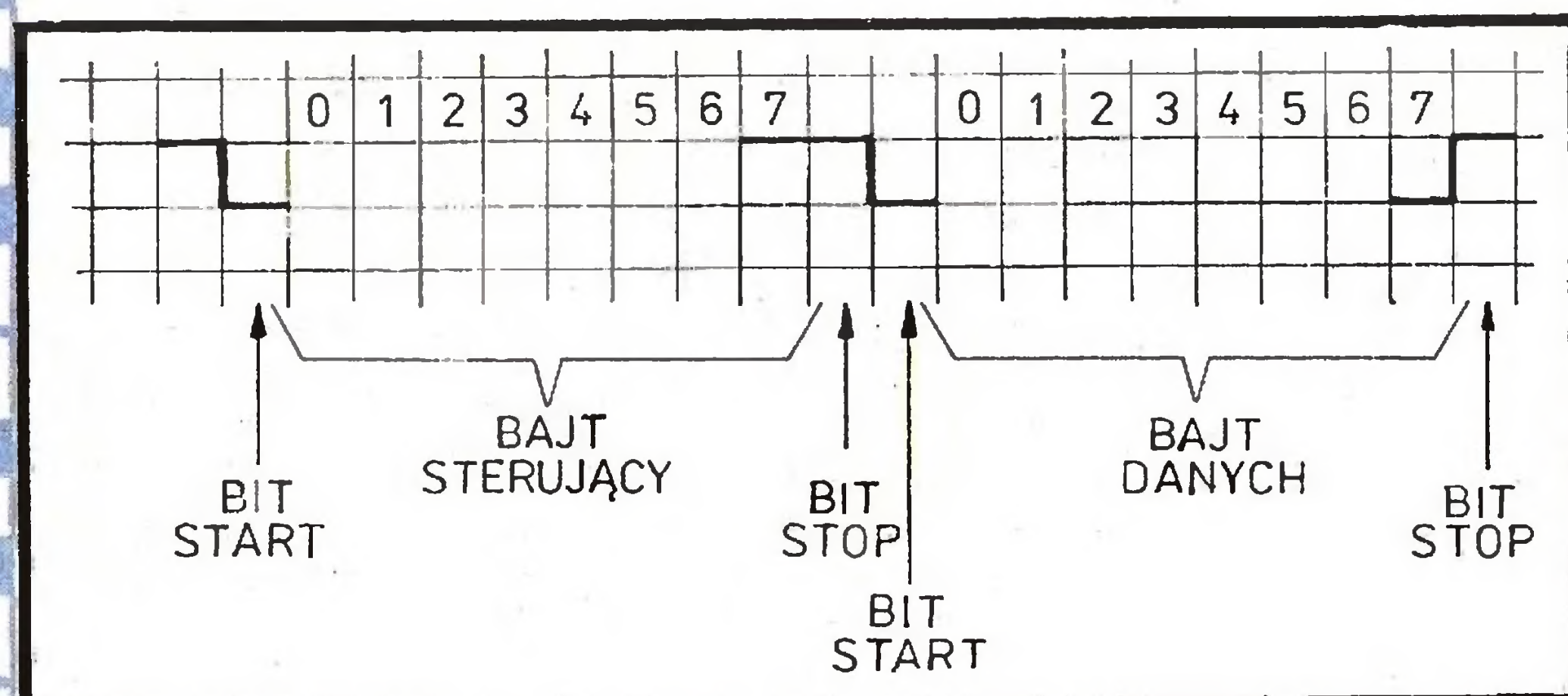
nuta	oktawa	kod	nuta	oktawa	kod	nuta	oktawa	kod
c	0	0	g#	3	44	e	7	88
c#	0	1	a	3	45	f	7	89
d	0	2	a#	3	46	f#	7	90
d#	0	3	h	3	47	g	7	91
e	0	4	c	4	48	g#	7	92
f	0	5	c#	4	49	a	7	93
f#	0	6	d	4	50	a#	7	94
g	0	7	d#	4	51	h	7	95
g#	0	8	e	4	52	c	8	96
a	0	9	f	4	53	c#	8	97
a#	0	10	f#	4	54	d	8	98
h	0	11	g	4	55	d#	8	99
c	1	12	g#	4	56	e	8	100
c#	1	13	a	4	57	f	8	101
d	1	14	a#	4	58	f#	8	102
d#	1	15	h	4	59	g	8	103
e	1	16	*c	5	60	g#	8	104
f	1	17	c#	5	61	a	8	105
f#	1	18	d	5	62	a#	8	106
g	1	19	d#	5	63	h	8	107
g#	1	20	e	5	64	c	9	108
a	1	21	f	5	65	c#	9	109
a#	1	22	f#	5	66	d	9	110
h	1	23	g	5	67	d#	9	111
c	2	24	g#	5	68	e	9	112
c#	2	25	a	5	69	f	9	113
d	2	26	a#	5	70	f#	9	114
d#	2	27	h	5	71	g	9	115
e	2	28	c	6	72	g#	9	116
f	2	29	c#	6	73	a	9	117
f#	2	30	d	6	74	a#	9	118
g	2	31	d#	6	75	h	9	119
g#	2	32	e	6	76	c	10	120
a	2	33	f	6	77	c#	10	121
a#	2	34	f#	6	78	d	10	122
h	2	35	g	6	79	d#	10	123
c	3	36	g#	6	80	e	10	124
c#	3	37	a	6	81	f	10	125
d	3	38	a#	6	82	f#	10	126
d#	3	39	h	6	83	g	10	127
e	3	40	c	7	84			
f	3	41	c#	7	85			
f#	3	42	d	7	86			
g	3	43	d#	7	87			

*środkowe 'C' na fortepianie

Tablica 1

liczba		funkcja
dec	hex	
0	\$00	Undefined
1	\$01	Modulation wheel or lever
2	\$02	Breath Controller
3	\$03	Undefined
4	\$04	Foot controller
5	\$05	Portamento time
6	\$06	Data entry MSB
7	\$07	Main volume
8	\$08	Balance
9	\$09	Undefined
10	\$0A	Pan
11	\$0B	Expression Controller
12-15	\$0C-\$0F	Undefined
16-19	\$10-\$13	General Purpose Controllers (#'s 1-4)
20-31	\$14-\$1F	Undefined
32-63	\$20-\$3F	LSB for values 0-31
64	\$40	Damper pedal (sustain)
65	\$41	Portamento
66	\$42	Sostenuto
67	\$43	Soft pedal
68	\$44	Undefined
69	\$45	Hold 2
70-79	\$46-\$4F	Undefined
80-83	\$50-\$53	General Purpose Controllers (#'s 5-8)
84-90	\$54-\$5A	Undefined
91	\$5B	External Effects Depth
92	\$5C	Tremolo Depth
93	\$5D	Chorus Depth
94	\$5E	Celeste (Detune) Depth
95	\$5F	Phaser Depth
96	\$60	Data increment
97	\$61	Data decrement
98	\$62	Non-Registered Parameter Number LSB
99	\$63	Non-Registered Parameter Number MSB
100	\$64	Registered Parameter Number LSB
101	\$65	Registered Parameter Number MSB
102-120	\$66-\$78	Undefined
121-127	\$79-\$7F	Reserved for Channel Mode Messages

Tablica 2



Rys. 1

1	0	1	- CHANNEL PRESSURE
1	1	0	- PITCH WHEEL CHANGE
1	1	1	- SYSTEM COMMON MESSAGE

Siedem pierwszych komunikatów to komunikaty „kanałowe” (CHANNEL MESSAGES) — adresowane do konkretnych kanałów; ostatni otwiera całą klasę komunikatów „systemowych” (SYSTEM COMMON MESSAGES) przeznaczonych dla wszystkich urządzeń pracujących w danej sieci **MIDI**.

W tym miejscu zapraszam wszystkich zainteresowanych czytelników do przestudiowania tabeli 3, ujmującej w syntetycznej formie wszystkie komunikaty przewidziane przez standard **MIDI**.

Komunikaty kanałowe — CHANNEL VOICE MESSAGES

Komunikaty kanałowe przekazują informację o wszelkich zdarzeniach mających miejsce w synteźatorze. Wśród nich, niewątpliwie, najważniejsze to **NOTE OFF** (klawisz zwolniony) i **NOTE ON** (klawisz wciśnięty). Po tych komunikatach następuje bajt danych określający wysokość dźwięku (patrz tablica 1), a następnie — w wypadku klawiatury wrażliwej na siłę uderzenia — bajt danych, który określa w skali od 0 do 127 dynamikę tego dźwięku.

Komunikat **PROGRAM CHANGE** (zmiana programu) jest wysyłany w przypadku zmiany programu synteźatora, czyli zestawu parametrów określających brzmienie. Po tym komunikacie wysyłany jest bajt danych zawierający numer aktualnego programu (od 0 do 127).

Komunikat **PITCH WHEEL CHANGE** informuje o zmianie położenia specjalnego manipulatora pozwalającego na płynną zmianę wysokości dźwięku. Jest to jedno z ważniejszych urządzeń w profesjonalnym synteźatorze. Następujące po tym komunikacie dane mogą mieć długość dwóch bajtów (w bardziej rozbudowanych synteźatorach), umożliwiając tym samym rozpoznawanie położenia tego manipulatora z dokładnością 14-bitową. Proste synteźatory, jak na przykład Casio CZ101, wykorzystują do tego celu tylko jeden (młodszy) bajt danych.

Użycie jakiegokolwiek innego spośród wymienionych manipulatorów synteźatora: przycisku, pokrętki, pedału itp., powoduje wystanie komunikatu **CONTROL CHANGE**. Po nim następują dwa bajty danych: pierwszy z nich określa numer użytego manipulatora (patrz tabela 2), drugi zaś jego 7-bitową wartość.

Komunikat **POLYPHONIC KEY PRESSURE** nadają rozbudowane modele synteźatorów, wyposażone w klawiaturę wrażliwą na siłę naciśnięcia klawisza. Przez łącze **MIDI** wysyłana jest informacja o sile naciśnięcia każdego z użytych klawiszy.

W przybliżeniu to samo można powiedzieć o komunikacie **CHANNEL PRESSURE**, z tą jednak różnicą, że bajt danych określa tylko siłę naciśnięcia najmocniej przyciśniętego klawisza.

Komunikaty przyporządkowania — CHANNEL MODE MESSAGES

Jak wynika z tabeli 3, jest to podzbiór komunikatów **CONTROL CHANGE** i określa on sześć różnych sposobów przyporządkowania urządzeń w sieci **MIDI**:

LOCAL CONTROL OFF — po odebraniu tego komunikatu synteźator podporządkowuje się całkowicie rozkazom z sieci **MIDI**. Jego manipulatory i przyciski nie mają żadnego wpływu na produkowany dźwięk;

LOCAL CONTROL ON — ten komunikat przywraca możliwość wpły-

wania na brzmienie dźwięku z konsoli synteźatora odbierającego dane **MIDI**;

ALL NOTES OFF — wycisza dźwięk we wszystkich kanałach jednocześnie;

OMNI ON/POLY ON — synteźator akceptuje komunikaty kanałowe niezależne od numeru kanału i przypisuje je kolejnym głosom w synteźatorze. Dźwięk produkowany jest polifonicznie;

OMNI ON/MONO ON — jak poprzednio synteźator, akceptuje wszystkie komunikaty kanałowe, lecz dźwięk generowany jest jednogłosowo (monofonicznie);

OMNI OFF/POLY ON — ignorowane są wszystkie komunikaty kanałowe, z wyjątkiem wybranego wcześniej kanału. Nadchodzące komunikaty są rozsyłane polifonicznie do kolejnych głosów synteźatora;

OMNI OFF/MONO ON — akceptowane komunikaty odbierane są z kanałów od ustawionego wcześniej do N (gdzie N jest liczbą głosów synteźatora) i przypisywane monofonicznie do głosów od 1 do N-1.

Komunikaty systemowe — SYSTEM COMMON MESSAGES

Są to komunikaty przeznaczone głównie dla sekwencerów i automatów perkusyjnych — umożliwiają synchronizację czasową tych urządzeń;

SONG POSITION POINTER — po odebraniu tego komunikatu sekwencer ustawia swój wewnętrzny licznik taktujący na wartość przekazaną w dwóch bajtach danych;

SONG SELECT — komunikat umożliwiający wybranie jednego spośród kilku utworów zarejestrowanych wcześniej w pamięci sekwencera. Bajt danych określa numer żadanego utworu;

TUNE REQUEST — jednobajtowy komunikat przeznaczony dla synteźatorów analogicznych, nakazujący dostrojenie ich oscylatorów.

Komunikaty sterowania w czasie rzeczywistym — SYSTEM REAL-TIME MESSAGES

Wykorzystywane do synchronizacji czasowej wszystkich urządzeń typu sekwencerów i automatów taktujących w danej sieci **MIDI**:

MIDI CLOCK — jest wysyłany przez sekwencer lub automat perkusyjny 24 razy w czasie trwania jednej ćwierćnoty;

START — nakazuje rozpoczęcie pracy wszystkich sekwencerów w sieci (ich liczniki wewnętrzne są zerowane);

STOP — wstrzymuje pracę sekwencerów i wyłącza wszystkie źródła dźwięku;

CONTINUE — jest używany w połączeniu z **SONG POSITION POINTER** do wznowiania pracy sekwencerów w określonym momencie;

ACTIVE SENSING — komunikat generowany i możliwy do odebrania tylko przez niektóre modele synteźatorów — podtrzymuje stan gotowości urządzenia do pracy w sieci w przypadku, gdy nie są nadawane komunikaty **MIDI**. Komunikat **ACTIVE SENSING** pojawia się wtedy raz na 300 ms, a jeśli i on zniknie, urządzenie zdolne do jego odebrania ustawia się w stan taki jak po włączeniu zasilania (reset);

SYSTEM RESET — nakazuje powrót wszystkich urządzeń w sieci do stanu początkowego — wszystkie uprzednie ustawienia są kasowane.

Komunikaty specjalne — SYSTEM EXCLUSIVE MESSAGES

bajt sterujący		bajt danych	
high nibble	low nibble		
CHANNEL VOICE MESSAGES			
Note Off	1000	Channel: 0-15	Note: 0-127
Note On	1001	Channel: 0-15	Velocity: 0-127
Polyphonic Key Pressure	1010	Channel: 0-15	Note: 0-127
Control Change	1011	Channel: 0-15	Ctrl Number: 0-127
Program Change	1101	Channel: 0-15	Program: 0-127
Channel Pressure	1110	Channel: 0-15	Pressure: 0-127
Pitch Wheel Chan-e	1111	Channel: 0-15	LSB: 0-127
CHANNEL MODE MESSAGES			
Channel Mode Change	1011	Channel: 0-15	Local Control: 122
			On: 0
			Off: 127
			All Notes Off: 123
			Dummy: 0
			Omni Mode Off: 124
			Dummy: 0
			Omni Mode On : 125
			Dummy: 0
			Mono Mode On : 126
			# Voices in Receiver
			Poly Mode On : 127
			Dummy: 0
SYSTEM COMMON MESSAGES			
Undefined	1111	0001	
Song Position Pointer	1111	0010	LSB: 0-127
Song Select	1111	0011	MSB: 0-127
Undefined	1111	0100	
Undefined	1111	0101	
Tune Request	1111	0111	
End of Exclusive	1111	0111	
SYSTEM REAL-TIME MESSAGES			
MIDI Clock	1111	1000	(24 beats per quarter note)
Undefined	1111	1001	
Start Sequence	1111	1010	
Continue Sequence	1111	1011	
Stop Sequence	1111	1101	
Active Sensing	1111	1110	(send once/300ms. when no other activity)
System Reset	1111	1111	
SYSTEM EXCLUSIVE MESSAGES			
Whatever	1111	0000	Manuf. ID: 0-127 (followed by data)
			End with EOX (11110111)

Tablica 3

Te komunikaty przeznaczone są dla specjalnych urządzeń **MIDI**, a mianowicie dla urządzeń, w których **MIDI** jest używane w zasadzie głównie do transmisji danych w celu ich dalszej obróbki, np. na komputerze. Po kodzie sterującym **SYSTEM EXCLUSIVE** wysyłany jest 7-bitowy kod identyfikacyjny (przydzielany przez **INTERNATIONAL MIDI ASSOCIATION**), po nim zaś dowolnie długi ciąg danych. Przesyłanie danych kończy komunikat **END OF EXCLUSIVE**, przerywa przesyłanie danych — dowolny bajt sterujący.

W tym miejscu można by zakończyć opisywanie protokołu **MIDI**, gdyby nie fakt, że **MIDI** bezustannie ewoluuje. Przesyłanie danych w trybie **SYSTEM EXCLUSIVE** jest w tej chwili najbardziej obiecującym kierunkiem rozwoju **MIDI**, gdyż w połączeniu z dynamicznie powstającym oprogramowaniem komputerów domowych umożliwia operacje, które do niedawna dały się zrealizować jedynie na wyspecjalizowanych urządzeniach. Mam tu na myśli przekazywanie danych z tzw. samplerów, czyli przetworników analogowo-cyfrowych do komputerów, celem ich dalszej obróbki lub analizy. Zasady transmisji danych łączem **MIDI** zostały ujęte w normę w czerwcu 1985 roku i noszą nazwę **SAMPLE DUMP STANDARD**.

Inny rodzaj informacji przekazywanej w trybie **SYSTEM EXCLUSIVE** to sygnały czasowe służące do precyzyjnej synchronizacji w czasie obrazu i dźwięku — wprowadzono je po to, aby ułatwić pracę muzykom i kompozytorom tworzącym podkłady dźwiękowe do filmów. Reguły kodowania czasu zostały również ujęte w odrębną normę — **MIDI TIME CODE**.

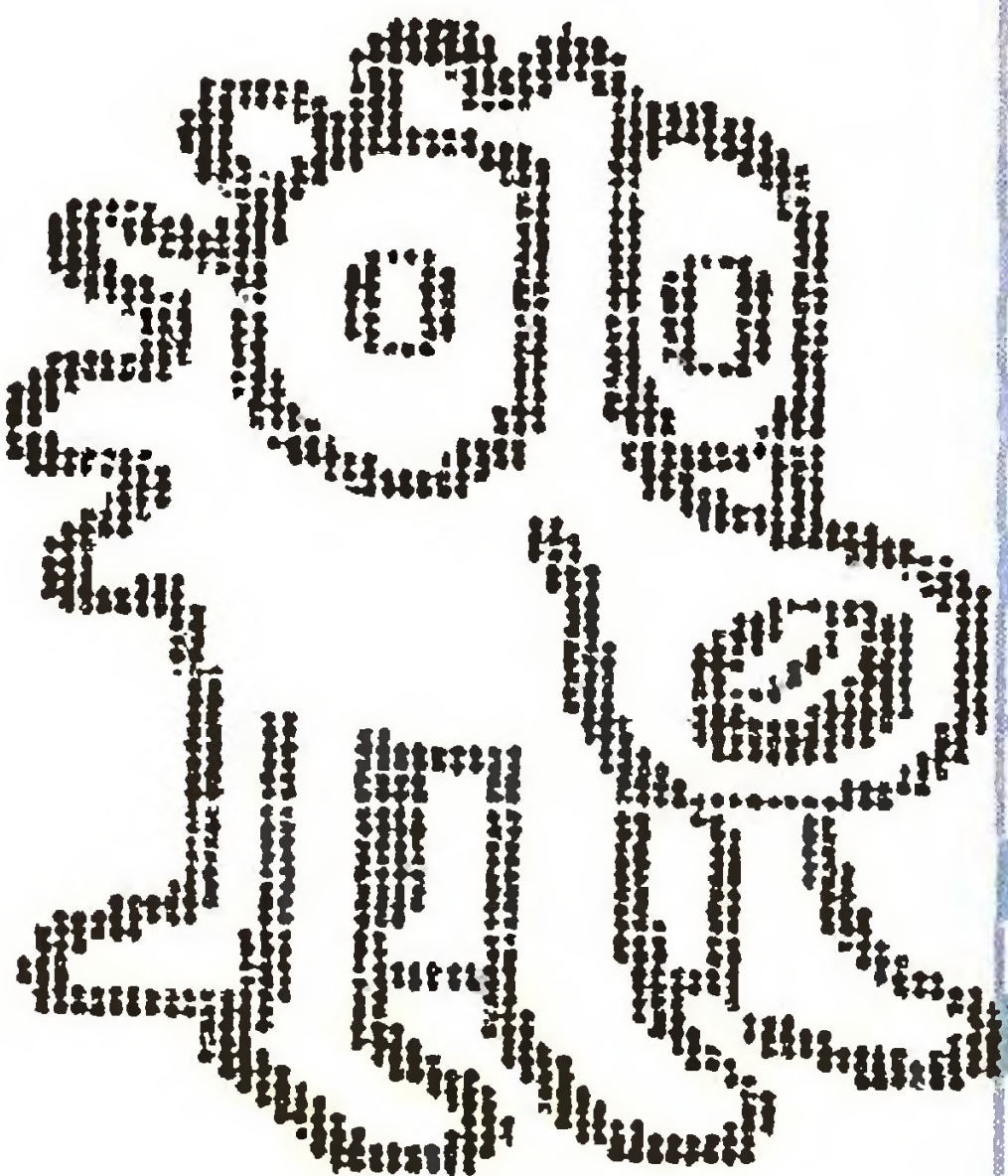
Jak więc widzicie, temat **MIDI** nie jest tak bardzo skomplikowany, ale przyznać trzeba, że po siedmiu latach samodzielnego życia i rozwoju jest to

dziedzina bardzo obszerna i owocująca bezustannie nowymi pomysłami.

Także w dziedzinie nas interesującej, czyli **MIDI** i komputery, każdy dosłownie miesiąc przynosi nowe oprogramowanie. Po programach symulujących na komputerze wielośladowe magnetofony, edytorach brzmień dla synteźatorów, edytorach partytur czy edytorach do obróbki danych z samplerów pojawiły się ostatnio programy symulujące profesjonalną konsolę mikserską.

W następnym odcinku tego cyklu postaram się opisać najbardziej interesujące oprogramowanie **MIDI** dla komputera Atari ST. Tymczasem wszystkich „nawiedzonych” przez **MIDI** posiadaczy 8-bitowego Atari zachęcam do zbudowania interfejsu **MIDI** do tego komputera i do pisania programów wykorzystujących taką przystawkę. Poprzez takie urządzenie mogłyby się komunikować — nie tylko muzycznie — małe Atari z Atari ST.

Tadeusz Pękacz



Atari PC4



Zainstalowaną w PC4 stację 3,5-calowych dysków elastycznych cechuje dwukrotnie większa pojemność, która wynosi 1,44 MB. Do tego dochodzi twardy dysk 60 MB lub dysk wymienny 44 MB, w najbardziej rozbudowanej wersji oba razem.

Klawiatura w pełni spełnia wymogi obowiązującego obecnie standardu komputerów AT. Jest ona niezawodna i ergonomiczna. Klawisze funkcyjne umieszczono w górnym rzędzie, wyodrębniono osobny blok klawiszy służących do przemieszczania kursora. Unowocześnienia te znacznie poprawiają komfort pracy, szczególnie przy programach służących do przetwarzania tekstów.

W standardowym wyposażeniu komputera znajduje się 14-calowy, monochromatyczny monitor Atari PCM 124. Praca w trybach CGA, VGA, MDA i Hercules jest możliwa bez instalowania dodatkowych kart rozszerzających. Rozdzielczość osiągnięta w poszczególnych trybach wynosi: 720 x 348 punktów w trybie karty MDA lub Hercules, 640 x 350 w trybie EGA lub 640 x 200 w CGA. Jeśli kogoś nie zadowala monitor monochromatyczny, to z łatwością można zastąpić go monitorem kolorowym.

Janusz Jarmoch

Na ubiegłorocznych targach CeBIT oraz na Targach Lipskich w stoiskach firmy Atari królował nowy model komputera, 16-bitowy Atari PC4. Jest on unowocześnioną wersją oferowanego już od ponad dwóch lat Atari PC2 i PC3.

Twórcy PC4 postawili sobie za cel opracowanie komputera w pełni kompatybilnego z AT, spełniającego wymagania nawet najbardziej wymagających użytkowników, którzy korzystają z najnowocześniejszego oprogramowania graficznego i do obróbki tekstów, przydatnego do profesjonalnych zastosowań w laboratoriach naukowo-badawczych oraz w przemyśle. Czy im się to udało, pokaże czas.

PC4 różni się od poprzednika nie tylko obudową jednostki centralnej. Zastosowano w nim bardzo szybki procesor 80286 ze standardową częstotliwością 8 lub 12 MHz (w PC2 z mikroprocesorem 8088 była możliwa praca z częstotliwością 4.77/8 MHz). Na życzenie oferowany jest również komputer w wykonaniu specjalnym 16 MHz, z zainstalowanym dodatkowo koprocesorem matematycznym 80287. Rezultatem tego jest olbrzymie

przyspieszenie szybkości pracy, które w efekcie daje znaczną oszczędność czasu, szczególnie istotną przy bardzo złożonych programach, przetwarzających duże zbiory danych. Liczbę złącz umożliwiających dołączenie dodatkowych kart zwiększono z czterech do pięciu. Dzięki temu podłączenie komputera do sieci, zainstalowanie dodatkowej pamięci czy przekazywanie informacji za pomocą modemu nie stwarza najmniejszego problemu.

Atari PC4 znacznie przewyższa swojego przodka PC2 pod względem dostępnej pamięci operacyjnej. Zastosowano nowocześniejsze urządzenia pamięci masowych. Pojemność RAM wzrosła z 512 KB do 1 MB, z możliwością rozszerzenia do 8 MB.

DANE TECHNICZNE

- Procesor 80286
- Częstotliwość zegara 8, 12, 16 MHz
- Złącza dodatkowych kart: 4 złącza do kart AT, 1 do karty XT
- 1 MB pamięci RAM z możliwością rozszerzenia do 8 MB
- Stacja dysków elastycznych 3,5 cala o pojemności 1,44 MB
- Dysk wymienny 44 MB lub dysk stały 60 MB
- Tryby CGA, EGA, MDA, Hercules, VGA
- 1 złącze równoległe (Centronics)
- 2 złącza szeregowo (RS 232C)
- Gniazdo dla koprocesora 80287
- **Wyposażenie standardowe:**
 - mysz
 - oprogramowanie: MS DOS 3.3, GW-Basic, MS-Windows

UWAGA!

DWA NOWE CZASOPISMA KOMPUTEROWE JUŻ W KIOSKACH!

UWAGA!

MOJE **Atari**

tylko dla posiadaczy Atari

i

TOP SECRET

magazyn fanów gier komputerowych