

ATARI - COMMODORE - ZX SPECTRUM - AMSTRAD CPC - AMIGA - APPLE - ACORN - IBM

# RETROKOMP

Mikrokomputery wczoraj i dziś

1 3



W numerze:  
**STREFA**  
**NINTENDO**

ISSN 2450-5862



9 772450 586007 >



# RetroKomp 13

Prawdziwa Giełda  
Komputery MTU  
Video Technology VZ200  
Casio FX-700P  
Historia systemu CP/M  
Sharp PC-5000

## ZX SPECTRUM 19

Sinclair QL  
Zastosowanie funkcji w języku Basic

## AMSTRAD CPC 23

Wypełnianie w Logo

## COMMODORE 25

Rodzina Commodore 264  
Przyspieszamy PET-a  
Simons' Basic i sprajty

## ATARI 36

Wykopaliska - Ninja  
Starball, czyli pinball w kosmosie  
Rozbudowa pamięci w małym Atari

## APPLE 44

Programowanie w edytorze tekstu  
Rozszerzenia graficzne  
Zmiana nazw poleceń DOS  
Cyfrowy termometr

## ACORN 53

RISC OS 3  
Komputery edukacyjne

## IBM PC 57

Ćwiczmy refleks i intelekt  
Dzielenie tekstu w IBM Basic  
Rozszerzenie GSX do języka CBasic

## AMIGA 65

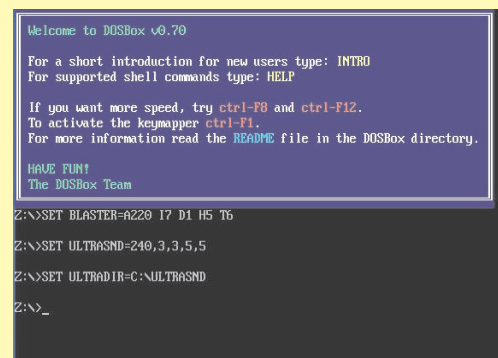
Zastępowanie czcionki Topaz  
Tworzenie i kompresja plików ADF

## NINTENDO 71

Gry przełomu roku 1990  
Nintendo Classic Mini

## ■ MS-DOS NA AMIDZE

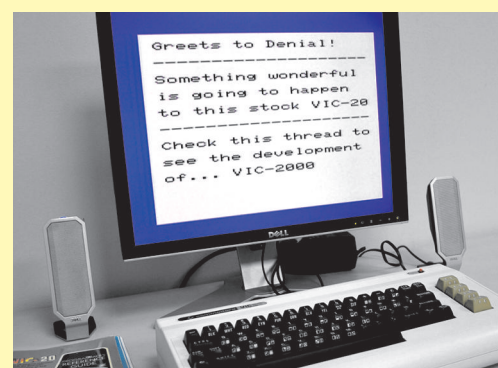
**DOS Box na Amigę.** Pojawiła się wersja tego emulatora dla klasycznego systemu Amigi, czyli dla procesorów serii 68k. Program posiada obsługę MIDI, co w przypadku Amigi przez wiele lat było problematyczne.



## ■ VIC EXPANDER

**Nowe rozszerzenie dla VIC-20.** Pojawiła się ciekawa możliwość rozbudowy tego modelu Commodore o wyjście wideo VGA, 4096 kolorów, 16 MB pamięci oraz 4-kanalowy dźwięk. Warto sprawdzić poniższą stronę:

<http://sleepingelephant.com/ipw-web/bulletin/bb/viewtopic.php?f=11&t=8734>:



## ■ C64 STUDIO

Jeśli chcecie pisać programy dla C64, pomoże Wam C64Studio. Jest to środowisko programistyczne współpracujące z emulatorem Vice. Można w nim pisać kod, a także wykonywać testy działania. Ostatnie poprawki obejmują modyfikacje edytora, menadżera plików, a także mechanizmu wyświetlania sprajtów. Więcej informacji:

<http://www.georgrottensteiner.de/en/index.html>

# Prawdziwa Giełda

ADAM ZALEPA

Ostatnio popularne staje się organizowanie imprez nawiązujących do giełd komputerowych odbywających się dawno temu. Niestety, w większości wypadków nie ma to zbyt wiele wspólnego ze specyfiką i klimatem starych giełd. W mojej rodzinnej Łodzi utrzymywały się one jeszcze nawet po roku 2000, choć cały czas zmieniały się dynamicznie. Jaka była stara Giełda?

Była przede wszystkim okazją i często jedną z niewielu możliwości dotknięcia komputerów oraz zdobycia oprogramowania. W niedawno nagrany filmie dokumentalnym na ten temat kilka osób mówi o tym, że dzięki giełdzie komputerowej powstała demo-scena. Wybaczenie, byłem stałym bywalcem na tych imprezach od lat '80-tych, a później nawet współorganizatorem w kilku miejscowościach w województwie łódzkim i mam zdecydowanie inne zdanie. Oczywiście, nie twierdzę, że ludzie się nie spotykali także czysto towarzysko. Z tych okazji wynikało z pewnością niejedno demo, moduł czy obrazek. Jednak był to tylko margines działalności giełd.

Podstawową sprawą było kopiowanie oprogramowania w czasach, gdy w Polsce jeszcze za to nie karano, a w Bajtku można było wydrukować swobodnie opinie mówiące o dwóch stronach medalu. Jeden obóz argumentował, że po unormowaniu rynku drastycznie spadnie dostępność do oprogramowania, a drugi - przedstawiał racje wydawców. Drugą najważniejszą sprawą był dostęp do sprzętu i... dyskietek. Nawet jeśli nie mogliśmy sobie pozwolić na zakup

nowej maszyny, świetnie było zobaczyć ją w akcji i porozmawiać z jej właścicielem, który zwykle był także pasjonatem swojego cacka. Natomiast dyskietki były potrzebne jeśli chcieliśmy zdobyć "nowości" lub po prostu przydatne programy albo gry. Dema były często odtwarzane na monitorach lub telewizorach, aby zareklamować swoje stoiska. Tak było na przykład ze słynnym polskim demem "Terminator" na Amigę. Większość osób pytało jednak o gry, nieco mniejsza grupa o programy, natomiast dema były na końcu typowej listy wyłożonej na stole. Nie chcę przez to w żaden sposób deprecjonować osiągnięć sceny, ale bądźmy wierni faktom.

Z pewnością nie zapomnę tego niepowtarzalnego klimatu prawdziwej Giełdy, gdy siedzieliśmy obok siebie, każdy wpatrzony w swój monitor i chwaliliśmy się swoją konfiguracją komputera. Okrzyki takie jak "040 rulez" były na porządku dziennym, a w momencie, gdy na giełdzie wystawiał się ktoś nowy z mało znanym sprzętem - często była to prawdziwa sensacja. To na jednej z giełd, choć już nie w Łodzi zobaczyłem demo Lemmingów na

C64 i usłyszałem pierwsze dźwięki z Amigi - z gry Rick Dangerous 2. To robiło wrażenie. Później imprezy zmieniły się nie do poznania - oprogramowanie zastąpił sprzęt, coraz mniej osób kopiowało dyskietki, a w ostatnich latach działalności giełdę można było określić mianem "aukcji dla zielonych". Sprzedawane produkty były w bardzo wysokich cenach i prawie nikt już nie handlował prywatnie - w większości były to oferty małych sklepów komputerowych. Później rynek zmienił się jeszcze bardziej i nawet te małe sklepy w większości przestały funkcjonować lub przyłączyły się do dużych sieci.

Dlatego jeśli mówimy o prawdziwym obrazie Giełdy, przedstawiamy jej różne oblicza, ale w odpowiednich proporcjach. Była to świetna możliwość zakupu, sprzedaży i wymiany wszystkiego, co dotyczyło komputerów, okazja do dyskusji na temat ulubionego sprzętu. Była to możliwość zarobku przez młodych ludzi wchodzących dopiero w dorosły świat. Pozostałe kwestie były dodatkiem - bardzo miłym, ale jednak tylko promilem w stosunku do całej reszty.

# MTU-130

MARCIN LIBICKI

**Micro Technology Unlimited to mało znana w Polsce amerykańska firma, która na samym początku lat '80-tych zbudowała komputer o nazwie MTU-130. Był oparty o popularny wtedy procesor 6502, a więc producent chciał zastosować sprawdzoną jednostkę w dość ciekawym zestawie. Mam na myśli stację dyskietek o pojemności od 500 KB aż do 1 MB.**

Komputer miał 80 KB pamięci RAM oraz stację dyskietek i 12-calowy monitor w fabrycznym zestawie. Według ówczesnej reklamy, miała to być maszyna zaprojektowana "luksusowo", aby użytkownik nie musiał zbyt szybko rozszerzać pamięci, ani kupować innych dodatków. Proponowano ceny od 4000 do nawet 5000 dolarów, co przy ok. 600 dolarów za wydany później Commodore 64 oparty na tym samym procesorze może wydawać się dość szokujące. Osobiście nie dziwię się, że przy takiej polityce firma Micro Technology przepadła w mroku dziejów. Ściślej mówiąc, firma działa nadal, ale prowadzi zupełnie innego typu działalność.

Wróćmy jednak do komputera MTU-130. Miał on kilka nowatorskich cech, na przykład procesor mógł adresować do 256 kilobajtów ciągłej pamięci. Dostępne były tryby graficzne od 240x256 pikseli (4 poziomy szarości) do 480x256 pikseli (czarno-biały), dający pełne 80 znaków w wierszu. Kolorów nie przewidziane, ponieważ dołączony monitor był "zielony", podobnie jak polskie Neptuny.

Całe oprogramowanie było wczytywane w pamięci RAM, co miało pozwolić na łatwo aktualizację w przyszłości, a także samodzielne modyfikacje dokonywane przez użytkownika. System operacyjny o nazwie CODOS pozwalał na współpracę między innymi napędów dyskietek, jak również drukarek igłowych o wysokiej wtedy rozdzielczości.

Ponadto komputer posiadał dwa porty równoległe i jeden port szeregowy RS-232 z programowalną szybkością transmisji danych. Jeśli chodzi o dźwięk komputer oferował ośmiobitowy przetwornik cyfrowo-analogowy z filtrem i wzmacniaczem.

Producent dołączył własny język programowania o nazwie MTU-Basic, chociaż w rzeczywistości był to rozszerzony dialekt wyprodukowany przez Microsoft. Ciekawostką jest fakt, że po włączeniu MTU-13 informuje o tym użytkownika syntetyzowanym głosem. Nie spotkałem się z takim rozwiązaniem w żadnym innym sprzęcie. Komunikat trwa tylko dwie sekundy i według dokumentacji zajmuje ok. 16 kilobajtów.

Firma Mico Technology reklamowała swój produkt także nawiązując do komputerów 16-bitowych. Pisano na przykład o możliwości zainstalowania drugiego, a nawet trzeciego procesora w formie Motoroli 68000. Chwalono się możliwością odczytywania dyskietek sformatowanych w systemach CP/M i MS-DOS. Podawano, że CODOS jest systemem z rodowodem uniksowym, a także przytaczano szybkość transmisji danych z dyskietki - wczytywanie 32 kilobajtów zajmować miało tylko 2 sekundy.

Szkoda, że dzisiaj praktyczne nigdzie nie można zobaczyć na żywo tego ciekawego sprzętu. Tym samym nie sprawdzimy, na ile reklama pokrywała się z rzeczywistością. Warto jednak odnotować, że na bazie słynnego procesora 6502 próbowano stworzyć komputer typu high-end z możliwością rozbudowy tak, aby z 8-bitowego komputera uzyskać jednostkę 16-bitową. Niestety, przy tym poziomie proponowanych cen musiało to skończyć się porażką. Slogan reklamowy "Change your view" musiał zostać zmieniony na "Change your computer".



# VZ-200

## Video Technology

MARCIN LIBICKI

**Video Technology VZ200 to kompaktowy mikrokomputer o dużych możliwościach i wielu ciekawych funkcjach, które były niedostępne w podobnym sprzęcie. Został wyprodukowany w 1983 roku i miał być podobny do słynnego ZX Spectrum.**

Maszyna jest oparta na procesorze 6502, czyli tym samym, który można znaleźć w komputerach Apple, Commodore i Atari. Posiada 12 KB pamięci ROM zawierającego monitor i specjalną implementację języka Microsoft Basic. Podstawowa pamięć RAM to tylko 4 KB, ale dostępne były moduły z dodatkową pamięcią 16 KB lub 64 KB. Były podłączane do gniazda za tyłu komputera, tak więc instalacja nie sprawiała użytkownikom problemów.

Dwie trzecie górnej części komputera zajmuje klawiatura z 45 klawiszami rozmieszczonych w czterech rzędach. Klawisze mają bardzo krótki skok, co przypomina mi ośbiście Timexa. W związku z tym pisanie bezwzrokowe nie jest najłatwiejsze. Odstępy między klawiszami są typowe, ale pewnym problemem jest klawisz Spacji. Jest on umieszczony w prawym dolnym rogu i ma wielkość klawisza Shift, który znajduje się bezpośrednio po drugiej stronie klawiatury.

W związku z powyższym, dostępny jest tylko jeden klawisz Shift, a dodatkowo znak zapytania jest umieszczony na klawiszu z literą L.

### Klawiatura

Jest w stylu "Chiclet" i klawisze mają bardzo krótki skok. Pisanie dotykowe jest możliwe tylko w dość ograniczony sposób. Chociaż odstępy między klawiszami są takie same jak w zwykłej maszynie do pisania, gumowane klawisze mają inne "wyczućcie".

Znacznie bardziej katastrofalne dla pisania dotykowego jest to, że nie ma spacji; zamiast tego klawisz spacji znajduje się na prawym końcu dolnego rzędu obok okresu. Oznacza to również, że jest tylko jeden klawisz Shift (na lewym końcu dolnego rzędu). Kilka innych klawiszy nie posiada oczekiwanych znaków; na przykład znak zapytania znajduje się na klawiszu L.

Dużo lepszą wiadomością jest fakt, że każdy klawisz - oprócz wpisywania pojedynczej litery, cyfry lub



znaku specjalnego - posiada inne funkcje. Wszystkie podstawowe polecenia, słowa kluczowe i funkcje mogą być wywoływane za pomocą pojedynczego naciśnięcia klawisza, połączenia z klawiszem funkcyjnym lub naciśnięcia dodatkowego klawisza, gdy naciśnięte są już inne.

Ogólnie możliwe do wywoływania są różne kombinacje klawiszy i działa to w dużo bardziej rozbudowany sposób niż nowoczesne klawiatury. Każdy klawisz na VZ200 pozwala na wpisanie podstawowego słowa kluczowego - czyli tak jak w ZX Spectrum - oraz jeden lub dwa znaki graficzne.

Reasumując - klawisze posiadają aż pięć trybów pracy: dwa podstawowe słowa kluczowe, dwa symbole graficzne i znak alfanumeryczny. Wciśnięcie klawisza powoduje wyemitowanie krótkiego sygnału dźwiękowego, czyli znowu mamy tu podobieństwo do sprzętu Sinclaira.

## Grafika

Rozdzielczość VZ200 wynosi 256x192 pikseli jest dostępna jako siatka 32x16 bloków o rozmiarach 8x8 pikseli, natomiast rozdzielczość widoczna na ekranie startowym to tylko 256x128 pikseli. Jest to spowodowane próbą ujednolicenia wyświetlania w systemach telewizyjnych PAL i NTSC, a także ograniczoną ilością pamięci RAM. Możliwe jest jednak wykorzystanie dodatkowych linii za pomocą kodu maszynowego.

Mamy 64 kody znaków graficznych, które mogą być wyświetlane w następujących kolorach: magenta, czarny, czerwony, pomarańczowy,

pomarańczowy, żółty, zielony, cyjan i niebieski. W trybie grafiki wysokiej rozdzielczości poszczególne piksele mogą być używane na siatce 128x64 w każdym z ośmiu kolorów. Aby ustawić odpowiednią lokalizację na ekranie, używana jest komenda SET(x,y). Poza tym można korzystać z polecenia RESET(x,y), które "wyłącza" dowolny piksel, a także POINT(x,y) sprawdza, czy piksel jest aktywny, czy też nie.

Komputer ma tylko 2 KB pamięci wideo, jednak opracowano kilka nieoficjalnych modyfikacji, które zwiększyły ilość pamięci VRAM i pozwoliły na swobodne używanie trybu typu 256x192 pikseli w kolorach.

## Pełnoekranowy edytor

Edytor tego komputera to wielka zaleta w stosunku do innego sprzętu z

epoki. W celu edytowania linii programu, nie jest konieczne wywoływanie komendy EDIT lub podobnych. Linia jest po prostu wyświetlana - sama lub wraz z całym listingiem programu. Za pomocą czterech klawiszy kierunkowych znajdujących się w prawym dolnym rogu klawiatury, kursor jest przesuwany do znaku, który ma być zmieniony.

Trzeba jednak uważać, bo zmiana linii następuje po naciśnięciu klawisza Return, ale kursor musi być umieszczony na końcu linii. Inaczej zmiana nie będzie w pełni zastosowana.

Jednak klawisze kierunkowe kursora działają dopiero po naciśnięciu klawisza Control po lewej stronie. Znajduje się on bezpośrednio nad klawisze Shift i jest dużo mniejszy, przez co można bardzo łatwo pomylić się jeśli nie patrzymy na





klawiaturę. Trzeba do tego się przyzwyczaić. Drugi problem polega na tym, że po chwili bufor przypisany do edycji programu bardzo szybko się przepełnia i nie można wpisywać zbyt długiego programu. Jest to oczywiście spowodowane małą ilością pamięci, tak więc w podstawowej konfiguracji może być raczej tylko ciekawostką.

## Urządzenia peryferyjne

VZ200 posiada interfejs do podłączenia standardowego magnetofonu kasetowego, który działa z prędkością ok. 600 bps (czyli bitów na sekundę). Jest to wolniejsze niż w przypadku innych komputerów z podobnego okresu, które mają prędkość nawet do 2400 bps, jednak trzeba przyznać, że jest to wartość dwukrotnie większa niż komputery z samego początku lat '80-tych.

W praktyce program, który zajmuje cały obszar 4 KB pamięci wczytuje się ok. 1 minuty, a program o objętości 16 KB potrzebuje na załadowanie ok. 4 minut.

Komputer generuje sygnał video dla monitora i wyjście telewizyjne (RF). Kolejna ciekawostka to fakt, że możemy pracować w jednym z trzech trybów: tekstu, grafiki mieszanej i tekstu oraz grafiki wysokiej rozdzielczości.

W trybie tekstowym VZ200 wyświetla 16 wierszy po 32 znaki, które mogą być pokazywane w postaci zwykłego obrazu lub w odwróconych kolorach (inwersja). Standardowy zestaw znaków charakteryzuje się występowaniem tylko wielkich liter.



## Problemy

Niestety, VZ200 potrafi się dość często zawieszać. Klawiatura posiada klawisz Break, ale nie daje to użytkownikowi pełnej kontroli nad komputerem. Niezbyt dobrze działa polecenie LLIST, które powinno powodować wydruk tekstu na drukarce. W tym wypadku Basic po prostu się zawiesza. Problem potęguje łatwość wpisania słowa LLIST, ponieważ klawiatura ma tendencję do "odbijania".

Kolejną sprawą jest reset komputera, bowiem po zawieszeniu nie raz trzeba kilka razy włączyć jednostkę, aby pamięć została wyzerowana. Nie jestem do końca pewien, czy jest to problem stanu mojego egzemplarza, ale może to powodować zniecierpliwienie. Sprzęt zawiesza się również wtedy, gdy spróbujemy wywołać urządzenie peryferyjne, które aktualnie nie jest podłączone. W niektórych przypadkach VZ200 pokazuje

komunikat o błędzie, ale bardzo często po prostu brak jest reakcji na działanie użytkownika. Z tego wynika, że obsługa błędów w Basicu nie została przetestowana zbyt dobrze.

## Podsumowanie

Ogólnie rzecz biorąc, VZ200 jest wszechstronnym małym komputerem, oczywiście jak na czasy, w którym powstał. Mimo pewnych niedociągnięć implementacja Microsoft Basica spełnia bardzo dobrze swoje funkcje. Szczególnie warto podkreślić, że bardzo łatwo można wywoływać różne funkcje graficzne, co nie jest łatwe w każdym Basicu na komputerach 8-bitowych.

Z tego, co sprawdziłem komputer miał kosztować mniej niż 100 dolarów, tak więc mógł być bardzo dobrą alternatywą dla bardziej popularnego, ale też droższego sprzętu. Szkoda, że nie uzyskał większej popularności.

# CASIO

## FX-700P

KAMIL STOKOWSKI

Produkt firmy Casio FX-700P to bardzo ciekawe urządzenie. Z jednej strony jest programowalnym kalkulatorem, a z drugiej posiada wbudowany pełny interpreter języka Basic. Z tego względu był sprzedawany także pod nazwą PC-4. Wyprodukowano także zewnętrzne urządzenia rozszerzające jego możliwości, jak drukarka termiczna FP-12 oraz interfejs do podłączenia magnetofonu o symbolu FA-3.

W kalkulatorze został użyty procesor HD61913A01 z wewnętrzną pamięcią oraz zintegrowanym sterownikiem wyświetlacza LCD i klawiatury. Może się komunikować z zewnętrzną pamięcią i peryferiami poprzez 4-bitową magistralę.

Komputer jest naprawdę mały, dzisiaj powiedzielibyśmy, że jest typowym urządzeniem mobilnym. Jest zasilany za pomocą baterii, przy czym sama jednostka, jak i urządzenia zewnętrzne mają inny rodzaj baterii. Komputer wykorzystuje dwie baterie litowo-jonowe, których żywotność producent określał na około 2 lata lub 300 godzin ciągłej pracy. Interfejs magnetofonu wykorzystuje dwie baterie alkaliczne AA, o deklarowanej żywotności wynoszącej również dwa lata.

Drukarka termiczna wykorzystuje akumulatory, które po całkowitym naładowaniu drukują około 3000

linii. W zestawie znajduje się ładowarka, która w pełni ładuje baterie drukarki w ciągu około 15 godzin.

Oczywiście deklarowaną żywotność podaję jako ciekawostkę, ponieważ widać po tym, do jakiej grupy docelowej był kierowany model FX-700P. Można go dodatkowo wyposażać w zewnętrzny moduł pamięci o pojemności 4 KB oraz interfejs szeregowy RS-232, tak więc nie ma problemu z przenoszeniem danych na inne komputery retro.

### Klawiatura

Klawiatura Casio nie jest typowa, bowiem zwykle w tego typu sprzedzie jest ona podzielona na dwie grupy: wiersze skonfigurowane jako "wyjście", natomiast kolumny oznaczają "wejście". Konkretny wciśnięty klawisz łączy wybrany wiersz z wybraną kolumną, aby wywołać określoną funkcję. Casio FX-

700P nie zachowuje tej konwencji. Klawisze nie mają żadnego podziału na wiersze i kolumny. Zaletą takiego rozwiązania jest mniej skomplikowany port przeznaczony do obsługi określonej liczby klawiszy, natomiast wadą jest bardziej skomplikowana obsługa.

### Pamięć

Kolejna ciekawa rzecz do pamięć, ponieważ jej pojemność nie jest podana w normalny sposób w kilobajtach. Zamiast tego, oryginalna dokumentacja stwierdza coś w tym rodzaju:

**Model FX-700P ma 26 komórek pamięci (zmiennych), które pozwalają na wykonanie 1568 kroków programu. Maksymalna ilość pamięci może być rozszerzona do 222.**

Mówiąc bardziej technicznie, kalkulator zawiera dwa układy scalone pamięci RAM typu HD61914A. Podobny model PB-100 ma włączony tylko jeden układ scalony pamięci, przy czym można dodać drugi znajdujący się wewnątrz opcjonalnego modułu rozszerzenia OR-1. Niestety interpreter języka Basic nie obsługuje rozszerzenia pamięci.

W praktyce wspomniany "krok" programu jest odpowiednikiem jednego znaku lub słowa kluczowego. Na przykład, linia:

**30 PRINT N(I)**

wykonuje się w aż 8 krokach, bowiem dwa są przewidziane dla liczby 30, jeden dla słowa kluczowego PRINT, po jednym dla znaków w N(I) i jeden dla zakończenia linii. Z tego wynika, że w zależności od rodzaju programów, które piszemy, FX-700P może być uważany za posiadający około 2 KB pamięci dostępnych dla programów oraz danych użytkownika.

FX-700P ma możliwość przechowywania do dziesięciu programów jednocześnie. Są one przechowywane w obszarach o symbolach od P0 do P9, do których można mieć dostęp podobnie, jak w przypadku zapamiętywania zmiennych. Wstarczy nacisnąć biały klawisz S, a następnie cyfrę od 0 do 9.

W praktyce występuje tu wiele podobieństw w stosunku do pełnowymiarowego komputera osobistego. Oczywiście, klawiatura jest znacznie mniejsza i mniej wygodna. Część klawiatury numerycznej znajduje się po prawej stronie, natomiast



klawiatura alfabetyczna jest zgodna ze standardowym układem QWERTY, jednak jej układ jest prostokątny, a klawisze są zdecydowanie mniejsze niż klawisze na klawiaturze numerycznej. Tak więc pisanie jest utrudnione, szczególnie jeśli chcemy zrobić to szybko.

Każdy klawisz ma trzy funkcje - wprowadza literę, która jest na nim nadrukowana, słowo kluczowe lub symbol widoczny na czerwono powyżej. Jest jeszcze funkcja poniżej, tym razem nadrukowana na niebiesko. Aby wprowadzić alternatywną funkcję poza znakiem, przed naciśnięciem klawisza z literą należy nacisnąć czerwony klawisz S lub niebieski klawisz F.

Dwa klawisze strzałek przesuwają kursor. Wszystkie klawisze Clear (AC) i Delete (DEL) działają tak samo jak w zwykłym kalkulatorze. Przycisk STOP zatrzymuje wykonywanie programu, natomiast klawisz wykonania (EXE) działa jak uniwersalny klawisz

wywołujący funkcje lub wprowadzający dane. Klawisz MODE wybiera opcję zapisu lub uruchomienia programu, włączenia lub wyłączenia drukarki, włączenia lub wyłączenia funkcji śledzenia programu oraz obliczenia wyrażone w stopniach lub radianach.

## Wyświetlacz

Wyświetlacz nie należy do pozytywnych cech FX-700P. Co prawda jest bardzo czytelny i kąt widzenia elementów LCD jest zaskakująco dobry, ale pokazuje tylko 12 znaków. Podczas pisania programów z dwucyfrowymi numerami linii, mamy tylko dziewięć znaków przeznaczonych dla instrukcji programu (pomijam tu znaki Spacji). Jest funkcja przewijania zawartości także poziomo, a linie mogą mieć do 62 znaków długości i mogą zawierać wiele różnych instrukcji.

Jednak na raz widać tylko 12 znaków, tak więc nie jest łatwo nie



tylko wprowadzać program, ale także porównywać poszczególne linie i poprawiać błędy.

## Basic

Każda zmienna (za wyjątkiem zmiennej typu ciąg) zajmuje 8 bajtów w pamięci RAM. Słowo najmniej znaczące umieszczane jest pod niższym adresem. Wartość "zerowa" reprezentuje zarówno zmienną numeryczną 0 jak i pusty ciąg. Dlatego poniższe linie są równoznaczne:

```
A$ = 0  
A$ = ""
```

Zmienna typu ciąg może przechowywać tylko do 7 znaków, natomiast zmienne numeryczne są przechowywane jako wartości zmiennoprzecinkowe. Pierwsze dwa 4-bitowe słowa zawierają dwie ostatnie cyfry wykładnika, przy czym najpierw zapisywana jest cyfra mniej

znacząca. Wykładnik zapisany jest w kodzie z przesunięciem, czyli do wartości wykładnika dodane jest przesunięcie o wartości 100, dzięki czemu wynik jest zawsze liczbą dodatnią. Trzecie 4-bitowe słowo zawiera pierwszą cyfrę wykładnika.

Dostępna jest również specjalna zmienna "\$" (również typu ciąg), która może przechować do 30 znaków. Tekst w pamięci kalkulatora poprzedzany jest liczbą oznaczającą długość zmiennej, a kończy się znacznikiem o wartości 255 (szesnastkowo \$FF). Oto przykład definicji zmiennej specjalnej:

```
$ = "Android"
```

Wiersze programu w Basicu rozpoczynają się numerem linii, który zapisywany jest jako 2 bajty i także kończy się wartością \$FF. Słowa kluczowe są przechowywane za pomocą jednobajtowych kodów, a

wartości numeryczne zapisane jako ciąg znaków oraz dwukropki rozdzielające kilka poleceń Basicu - w jednym wierszu jako wartość \$FE. Pliki tworzone przez Casio składają się z nazwy, a następnie z jednego lub kilku segmentów danych. Każdy z nich rozpoczyna się następującymi bajtami:

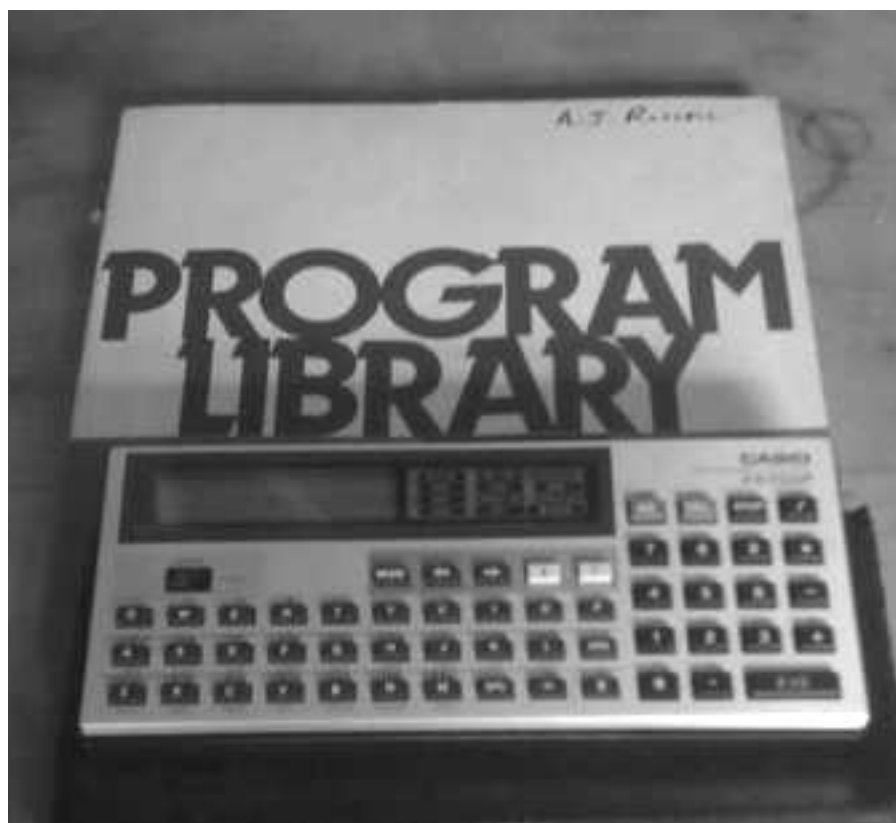
**\$DX** - oznacza plik zapisany za pomocą polecenia SAVE,

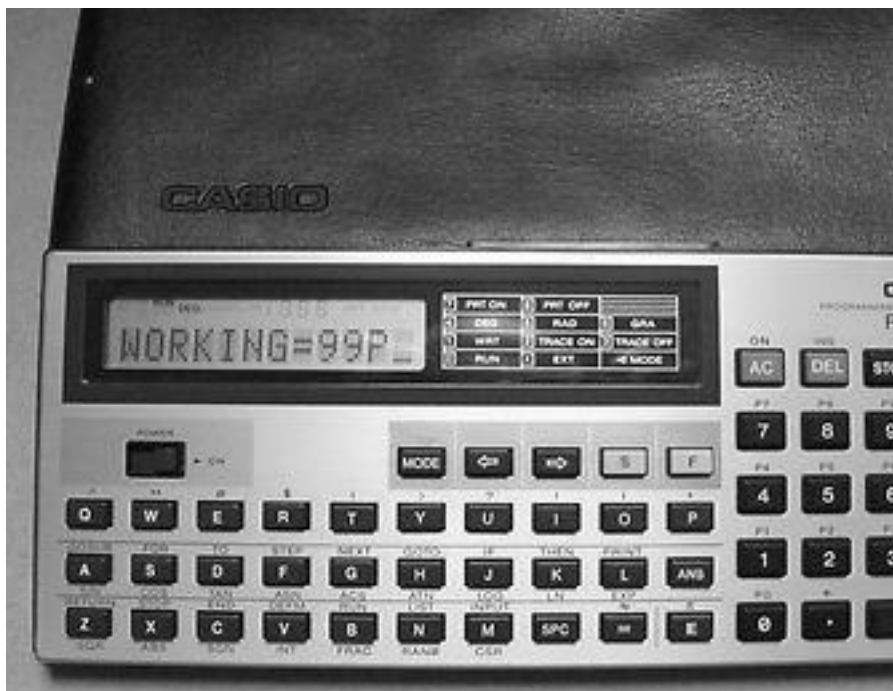
**\$EX** - plik utworzony za pomocą instrukcji PUT,

**\$FX** - plik utworzony za pomocą instrukcji SAVE A

gdzie "X" jest długością nazwy pliku - do 8 znaków). Plik może też nie posiadać nazwy, co jest kolejną interesującą cechą, ale w tym wypadku otrzymamy bajt \$F. Kolejnych 8 bajtów zawiera znaki nazwy pliku, natomiast ostatnie 2 bajty zawierają numer pierwszego wiersza programu w języku Basic zapisanego poleceniem SAVE. Segmenty danych rozpoczynają się bajtem \$02, a kończą - \$F0 (ostatni segment) lub \$F1. Ilość bajtów danych pomiędzy nimi nie może przekroczyć liczby 63, ponieważ zawartość jest wczytywana do 64-bajтового bufora. To ograniczenie nie dotyczy plików zapisanych przez polecenie SAVE A, który zawiera tylko pojedynczy segment danych.

Plik zawierający program w Basicu składa się z szeregu następujących po sobie wierszy. Jeżeli plik zostanie utworzony za pomocą polecenia SAVE A, każdy program będzie zakończony bajtem \$E0. Pusty program zawiera tylko bajt \$E0. Plik zapisany poprzez instrukcję PUT składa się z wartości zmiennych rozdzielonych





podręcznych numerów telefonów. Pozwala na wpisanie do 50 nazwisk oraz numerów.

## Wnioski

FX-700P jest kompaktowy, lekki i na pewno w swoim czasie był bardzo ciekawą propozycją. Łączy możliwość prostego programowania w języku Basic z funkcjami zwykłego kalkulatora. Został wyprodukowany w 1983 roku i kosztował 200 dolarów, a więc nie był najtańszy, ale umożliwiał wykonywanie "w terenie" podstawowych operacji komputera domowego. Pod tym względem mógł być bardzo przydatny studentom lub inżynierom, a dzisiaj może być prawdziwym wyzwaniem dla członków demo-sceny. Skoro można pisać dema wyświetlające obrazy na oscyloskopie, być może Casio FX-700P poddałby się innym ciekawym zabiegom zdolnego koderka?

znacznikiem \$FF. Warto dodać, że nazwy zmiennych nie są zapisywane.

Dostępne funkcje nie różnią się od typowego dialektu Basica. Mamy możliwość używania słów takich jak SIN, COS, TAN, a także pierwiastki kwadratowe, wartości bezwzględne, ułamki, liczby losowe oraz zaokrąglania. Jak w przypadku większości kalkulatorów, obliczenia mogą być wykonywane bezpośrednio do wprowadzenia do pamięci. Jednak FX-700P wykorzystuje tryb będący połączeniem zwykłego kalkulatora z językiem programowania, dlatego pod tym względem można znacznie łatwiej definiować większą ilość zmiennych i wykonywać bardziej skomplikowane obliczenia między nimi.

Co prawda język Basic jest uproszczony, bo posiada tylko 21 poleceń. Jednak obecne są wszystkie podstawowe słowa kluczowe, z wyjątkiem pozycji związanych z grafiką i funkcją drukowania. Nie mamy poleceń typu PEEK i POKE, nie można też korzystać z pętli (GOTO), podprogramów

(GOSUB), jak również instrukcji warunkowych (IF...THEN...ELSE). Mimo to istnieje możliwość wprowadzania ciekawych programów, choć programista jest w dużym stopniu ograniczony. Programy są zapisywane za pomocą typowego polecenia SAVE i ładowane za pomocą LOAD. Można podać nazwę lub wpisać samą nazwę polecenia. Dane są zapisywane za pomocą PUT i odzyskiwane za pomocą GET, natomiast nazwy plików są opcjonalne. Dostępna jest także instrukcja VER, która pozwala sprawdzić poprawność zapisu programu.

Do urządzenia dołączony był pakiet oprogramowania o nazwie Multipac, napisany przez firmę Sunrise Software. Składa się z czterech programów, w tym arkusza kalkulacyjnego, ale ponownie jego użyteczność jest mocno ograniczona. Możemy korzystać maksymalnie z pięciu wierszy i takiej samej ilości kolumn. Innym programem jest Fone File, czyli system do przechowywania

## Protokół komunikacji z drukarką:

Kalkulator zapisuje drukowany znak do pamięci RAM do adresów \$0132-\$0133, a następnie wysyła komendę sterującą \$F, po czym zwalnia magistralę. Drukarka po wykryciu tej operacji przejmuje kontrolę nad magistralą i odczytuje dane z pamięci RAM, generując niezbędne sygnały sterujące. Jeżeli odczytany bajt jest równy \$FE, limit czasu na zwolnienie magistrali wynosi 300 ms, w przeciwnym razie 4 sekundy. Po wydrukowaniu znaku drukarka wysyła komendę sterującą \$E, po czym zwalnia magistralę. Kalkulator po wykryciu tej operacji przejmuje kontrolę nad magistralą i wysyła komendę sterującą \$E. Powyższe kroki są powtarzane, aż wszystkie znaki zostaną wydrukowane.

# CP/M

## Historia systemu

MARCIN LIBICKI

Tak popularny dziś system operacyjny CP/M ma swoje korzenie w czasach, gdy mikrokomputery jakie znamy dzisiaj były dopiero pieśnią przyszłości. Projektantem system był Gary Kildall, który na początku lat 70-tych XX pracował jako konsultant w firmie Intel. Na co dzień Kildall pracował jako profesor informatyki w Podyplomowej Szkole Morskiej w Monterey (USA). Jego dwa stanowiska pracy stawiały go w wyjątkowej pozycji do obserwowania rodzącego się rynku i tworzenia czegoś zupełnie nowego.

Gary zaczął zbierać elementy, które w 1973 roku mogłyby stworzyć domowy system komputerowy. Główny procesor i pamięć to oczywiście układy scalone firmy Intel. Potrzebując czegoś, co pozwoliłoby połączyć wszystkie te komponenty w spójną używalną całość, Kildall napisał prosty system operacyjny w swoim ulubionym wówczas języku, PL/M. Jest to strukturalny język programowania wysokiego poziomu, który umożliwia programowanie modularne, przy czym jedną z ważniejszych jego cech jest tworzenie modułów i możliwość programowania z wykorzystaniem kodu maszynowego.

Kildall rezultat swojej pracy nazwał "Control Program/Monitor", czy w skrócie CP/M. Biorąc pod uwagę nazwę, jest to więc zestaw oprogramowania, które steruje podsta-

wowymi komponentami komputera. Dzisiaj nazywamy to po prostu systemem operacyjnym. Autor pokazał swoje wczesne wersje CP/M w firmie, ale Intel odmówił wprowadzenia go na rynek, a także dalszego rozwoju projektu. Nie jest to zaskakujące, ponieważ w 1973 i 1974 roku komputery były rzadkością, a ludzie, którzy je mieli nie byli pewni, co dokładnie chcą z nimi zrobić. Traktowano je bardziej jako ciekawostkę lub zabawkę. Ponadto Intel poświęcił już wtedy sporo czasu na rozwijanie własnego systemu operacyjnego o nazwie ICIS.

Był on z kolei dyskowym systemem operacyjnym, który miał być przyszłością firmy Intel, a później dodano nawet opcję konwersji danych i programów tak, aby można je było uruchamiać w CP/M. ISIS był jednozadaniowy i składał się z zestawu

procedur umożliwiających w jednolity sposób traktować urządzenia zewnętrzne. Oznacza to możliwość realizacji w danym momencie tylko jednego zadania, bowiem wykonanie określonej operacji wstrzymuje wykonanie całego podprogramu, aż do zakończenia realizacji wywołanej procedury systemowej.

Do 1975 roku komputer można było kupić tylko w kilku firmach, które traktowały swoich klientów jak hobbystów. Większość z tych firm tworzyła komputery z napędami dyskowymi, zwykle tworzyły one również własny system operacyjny. Gdyby ci pionierzy - jak Altair, IMSAI, Polymorphic i Processor Technology - byli w stanie szybko dostarczyć swoje produkty na rynek masowy, być może CP/M nie stałby się później standardowym systemem operacyjnym, z którego wyewoluował także

MS-DOS. Jednak właściciele komputerów nie zamierzali czekać, aż firmy wyjdą im naprzeciw z nowymi produktami, więc kupowali cokolwiek, co mogli znaleźć i wydawało im się warte uwagi. Dlatego też firmy trzebie, takie jak NorthStar, Vector Graphic i Cromemco zaczęły sprzedawać produkty przeznaczone dla komputerów innych marek, jak choćby Altair.

Kilku producentów zdecydowało się wyeliminować kosztowny etap badań i rozwoju i przyjęło system operacyjny CP/M jako domyślne oprogramowanie dla swoich produktów. Wśród tych firm na szczególną uwagę zasługują Tarbell Electronics i Digital Microsystems, bowiem jako jedne z pierwszych dostarczyły one gotowe systemy wraz z napędami dyskowymi. Produkowały komponenty w formie rozszerzeń, a więc ich produkty mogły być stosowane praktycznie w każdym systemie. Dzięki temu użytkownicy nie musieli czekać, aż producenci ich komputerów będą w stanie wyprodukować własne napędy. Zdarzało się również tak, że obiecywano własny system, a później wprowadzano CP/M pod własną nazwą. Daleko nie trzeba szukać, podobny wariant przyjął przecież polski zakład Elwro dla komputerów Junior.

Kolejnym ważnym elementem w historii CP/M był entuzjazm jego pierwszych użytkowników. Ci prawdziwi hobbysci zajmowali się problemami pozornie nie do przewyciężenia, ale dzięki temu nabywali szybko wiedzę i nabywali doświadczenia. System CP/M teoretycznie może zostać uruchomiony na dowolnym komputerze z procesorem Intel 8088 lub Z80, może także pracować na dowolnym napędzie dyskowym. Spowodowało



to powstanie dużej grupy użytkowników łączących wspólne zainteresowania.

Piszę o tym, ponieważ nie można nie doceniać znaczenia wsparcia tej silnej grupy w początkowych latach funkcjonowania branży komputerowej. Pamiętajmy, że szczegółowe informacje o produktach nie były łatwo dostępne. Choć producenci często wydawali produkty z pełną dokumentacją, sklepy komputerowe działały różnie, dlatego w niektórych przypadkach środowisko użytkowników było bardziej stabilne niż firmy.

Gdy producenci zaczęli dostarczać dobrej jakości dyski twarde, twórcy oprogramowania zapoczątkowali kolejną istotną fazę ewolucji systemu CP/M. Szybko zorientowali się, że kluczem do tego, aby rozwój oprogramowania był finansowo opłacalny jest pisanie programów, które działają na jak największej liczbie różnych komputerów. CP/M był jednym z niewielu systemów opera-

cyjnych, które mogły działać na dowolnym komputerze, ograniczał to tylko procesor. Z tego względu był to najlepszy wybór dla programistów.

Na szczęście dla CP/M, pierwszymi programami, które stały się dostępne były narzędzia programistyczne. W gruncie rzeczy to one pomogły uczynić CP/M wiodącym systemem operacyjnym. Wśród tych programów znalazły się takie pozycje jak CBasic oraz jego poprzednik (EBasic), Microsoft Basic oraz kilka specjalnych programów pozwalających używać kodu maszynowego. Narzędzia te były wykorzystywane do pisania programów użytkowych, takich jak na przykład bazy danych czy edytory tekstu.

Popularność systemu operacyjnego CP/M stała się częścią większego schematu. System powodował powstawanie języków programowania i innych narzędzi developerskich, a one napędzały z kolei rozwój programów użytkowych. Doprowadzało



to znowu do tworzenia większej ilości narzędzi programistycznych. Ta spirala doprowadziła do ciągłego zwiększania sprzedaży zarówno oprogramowania, jak i sprzętu działającego w środowisku systemu CP/M.

Oczywiście, podczas tego szybkiego rozwoju, CP/M nie pozostawał taki sam jak w momencie jego pierwszego wydania. Oryginalny system, który napisał Kildall nosił nazwę CP/M wersja 1.3. Co ciekawe, współtwórczynią była żona Garego, Dorothy McEwen, a sam system był sprzedawany za pośrednictwem ich wspólnej firmy, czyli wszystkim dzisiaj znanej - Digital Research).

CP/M w wersji 1.3 miał kilka błędów, dlatego Kildall zmodyfikował kod, a także dostosował do niego niektóre dołączone programy użytkowe. W tej sposób powstała wersja 1.4. W tym samym okresie Digital Research sprzedała prawa do CP/M 1.3 firmie Cromemco, która dokonała własnych

modyfikacji, tworząc CDOS (Cromemco Disk Operating System).

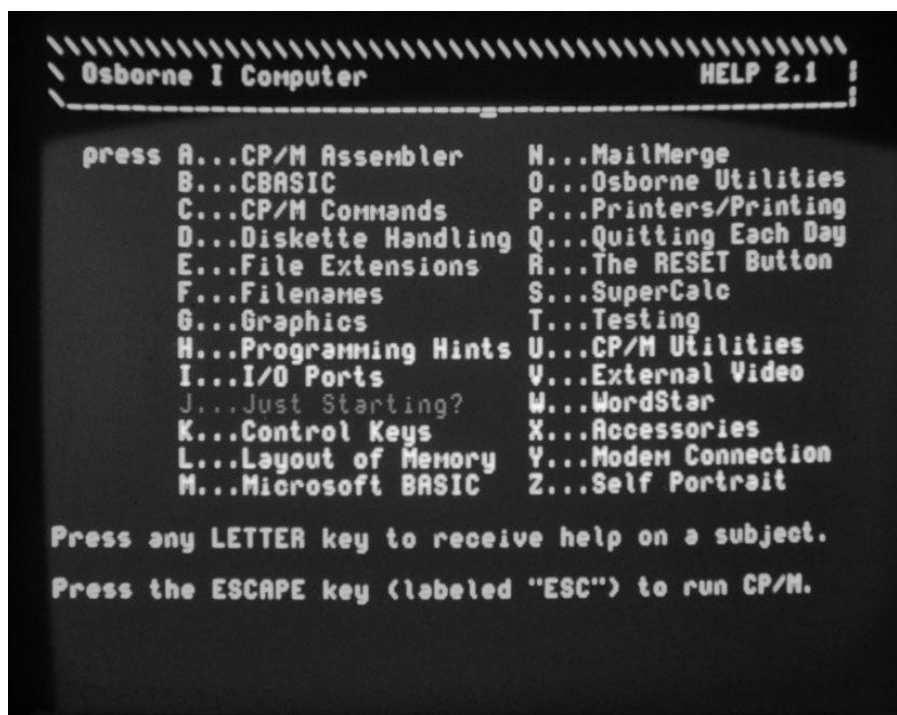
Warto dodać, że CP/M 1.4 został zaprojektowany w czasie, gdy pamięć była droga, a napędy dyskowe występowały tylko w jednej gęstości. Poza tym niezwykle popularne były również napędy taśmowe oraz konsole typu klawiatura z drukarką, które dzisiaj są praktycznie zupełnie zapomniane. W rezultacie, wydanie 1.4 zostało zaprojektowane z uwzględnieniem ograniczeń tych urządzeń. System zajmuje niewiele pamięci, co potęguje ograniczenia oprogramowania. Ogólnie rzecz biorąc jest uproszczony i został zaprojektowany tak, aby umożliwić rozbudowę oraz dołączanie kolejnych urządzeń peryferyjnych. Są to cechy, które sprawiły, że CP/M stał się popularny i ugruntował swoją pozycję jako standard. Nie był specyficzny dla konkretnego typu komputera i mógł pracować na minimalnej konfiguracji sprzętowej.

Komputery zmieniły się w 1979 i

1980 roku. Wraz z wprowadzeniem programów takich programów jak VisiCalc i WordStar, a także z nowego sprzętu jak dyski o większej pojemności i tańszej pamięci, komputery zaczęły być naprawdę popularne i przydatne nie tylko hobbystom. CP/M zmienił się wraz z komputerami. Wydanie 2.2 miało na celu aktualizację oryginalnego systemu, aby był bardziej zgodny ze zmieniającymi się komputerami. CP/M 1.4 był poważnie ograniczony pod względem ilości pamięci, natomiast wersja 2.2 została rozbudowana, aby umożliwić instalację aż do 16 napędów o pojemności już rzędu megabajtów.

Nastąpiły również inne zmiany. Większość z nich to drobne ulepszenia lub modyfikacje, które spowodowały, że CP/M stał się bardziej elastyczny. Dopuszczono możliwość jednoczesnego adresowania więcej niż jednego typu napędu dyskowego, na przykład można było korzystać z urządzeń o różnych parametrach, jak napędy 8-calowe i 5,25-calowe.

Jednak oryginalne cechy CP/M były nadal widoczne, system ciągle był kompaktowy i elastyczny. Jednak badania rynku wykonywane przez producenta koncentrowały się głównie na sprzedaży producentom i dystrybutorom, a nie użytkownikom. Dlatego firma Digital Research mogła sprzedawać dużą ilość oprogramowania przy stosunkowo niewielkim wysiłku. Najlepszym tego przykładem jest wprowadzenie przez Microsoft karty SoftCard Z80 dla komputera Apple II, do której dołączony był CP/M 2.2. Dzięki temu w ciągu kilku miesięcy pojawiło się kilkadziesiąt tysięcy nowych użytkowników systemu.





W latach 1979-1981 CP/M stał się na tyle popularny, że producent nie musiał praktycznie nic robić, aby utrzymać popularność. System przyciągnął firmy programistyczne, które pisały nowe oprogramowanie, co z kolei przyciągnęło się do przyciągnięcia większej ilości użytkowników. To z kolei powodowało zainteresowanie kolejnych producentów. Stało się to tak szybko, że w latach 1979-1983 ilość użytkowników zwiększyła się z ok. 300 tys. do 1,5 miliona.

Także rok 1981 był przełomowy dla CP/M. Wraz z wprowadzeniem nowych komputerów przez takie firmy-giganty jak IBM, Hewlett-Packard i Xerox, z których wszystkie oferowane były z systemem CP/M, liczba użytkowników systemu zaczęła rosnąć w przyspieszonym tempie. Jednak w tym samym roku szybko rozwijająca się technologia po raz kolejny spowodowała, że Kidall musiał rozpocząć prace nad udoskonaleniem oprogramowania.

CP/M został pierwotnie zaprojektowany do pracy na procesorach 8080. W 1981 roku nie była to najnowocześniejsza jednostka, dlatego wybrano Z80, który był nie tylko szybszy, ale także jego rozszerzony zestaw instrukcji pozwalał na lepszą optymalizację oprogramowania. To dlatego oryginalny CP/M może działać na komputerze opartym na układzie Z80 lub 8080.

Prędkość procesora Z80 zachęciła projektantów, aby stworzyć komputery obsługujące więcej niż jednego użytkownika jednocześnie. Ponadto, malejący koszt procesorów oznaczał, że projektant może po prostu dodać procesor do dyspozycji użytkownika. Tym samym popularne miały stać się komputery z obsługą wielu użytkowników. Jednak CP/M działa tylko w trybie jednego użytkownika, dlatego Digital Research rozpoczął rozwój drugiej gałęzi systemu, tworząc produkt o nazwie MP/M. Oznaczało to "Multiple Program/Monitor", choć nigdy nie zostało to oficjalnie potwierdzone.

MP/M działa podobnie jak CP/M, ale nie jest dokładnie taki sam. Oprogramowanie, które współpracuje z CP/M nie zawsze działa pod MP/M. Jedną z rzeczy, którą Digital Research przeniosło na nowy system był format dysku. Dysk utworzony na systemie MP/M jest bezpośrednio kompatybilny z systemem CP/M, oczywiście przy założeniu takiej samej gęstości i rozmiaru dysku. Oznacza to również, że dane i niektóre programy mogą być używane zamiennie w obu systemach.

Na początku 1983 roku firma Digital Research kontynuowała prace nad udoskonaleniem oryginalnej wersji systemu i rozszerzeniem CP/M na inne procesory. Dostępne stały się wersje dla procesorów Motorola 68000 i Zilog Z8000. Wynikiem tego jest produkt, który został nazwany CP/M-Plus. Jest to CP/M 2.2 z dodatkami podobnymi do tych, które można znaleźć w tym samym systemie przeznaczonym dla pozostałych procesorów.

Tak więc wraz z wprowadzeniem 16-bitowych procesorów pojawiła się kolejna gałąź CP/M. Pierwszym praktycznie dostępnym 16-bitowym procesorem był 8086 i jego kuzyn 8088. Digital Research planował wydanie nowej wersji CP/M, ale firma została uprzedzona przez Seattle Computer Products, która oferowała system podobny do CP/M. Nazwano go "86-DOS". Był on wzorowany na CP/M 1.4 z jedną bardzo ważną różnicą - zmieniony został format dysku, jak również zupełnie inna jest struktura danych. Później system został licencjonowany i zakupiony przez Microsoft, a dalej rozwijano go już w formie nowych produktów - MS-DOS i PC DOS.

# SHARP

## PC-5000

KAMIL STOKOWSKI

Na początku lat '80-tych firma Sharp zajmowała drugie miejsce w branży komputerowej, zaraz za liderem - firmą NEC. Jednocześnie nie prowadzono prawie żadnych działań marketingowych na terenie USA. Przyczyny takiego stanu rzeczy były dwie. Po pierwsze, ówczesne kanały dystrybucji to przede wszystkim sprzedawcy produktów biurowych, którzy w większości przypadków nie odnieśli sukcesu w sprzedaży komputerów. Po drugie, pierwsze wejście na rynek komputerowy firmy Sharp nie było udane ze względu na braki w oprogramowaniu.

W końcu firma stwierdziła, że produkt o oznaczeniu PC-5000 zmieni to wszystko. Różnił się znacznie od konkurencyjnych produktów, dlatego firma chciała to wykorzystać. Za cel postawiono uzyskanie dostępności nowego sprzętu w 800 sklepach komputerowych do końca 1984 roku. Co więcej, firma nie planowała dystrybucji za pośrednictwem wielkich sieci handlowych, co miało pomóc w uniknięciu wyniszczających wojen cenowych.

Co można powiedzieć o tym komputerze na pierwszy rzut oka? Z pewnością był bardzo nowatorską konstrukcją i można go traktować jako pionierską próbę stworzenia przenośnego komputera bez kompromisów dotyczących parametrów technicznych. PC-5000 był w dużej

mierze kompatybilny ze standardem IBM PC i miał procesor Intel 8088 o częstotliwości 4,77 MHz. Pamięć 128 KB pozwalała uruchomić system MS-DOS w wersji 2.0, który był zapisany w pamięci ROM. Pamięć można rozszerzyć, co jednocześnie zwiększało bazę dostępnego oprogramowania.

Komputer posiada wyświetlacz LCD o rozdzielczości 640×80 pikseli, co pozwala uzyskać 80 znaków, ale tylko 8 linii. Pełnoekranowa klawiatura oraz zewnętrzny napęd dyskietek 5,25-calowych dopełniały całość zestawu. Godną uwagi cechą komputera jest także wbudowana drukarka termiczna, którą można było również zakupić oddzielnie. Drukarka tego typu jest specyficznym rozwiązaniem, ale mogła być mała.

### Wnętrze

Mówiąc bardziej szczegółowo o specyfikacji technicznej, komputer PC-5000 ma 192 KB pamięci ROM, ale obszar 64 zarezerwowany jest dla systemu operacyjnego. Dlatego tylko 128 KB jest dostępnych dla MS-DOS oraz - co ciekawe - wbudowanego języka Basic. Wewnątrz znajduje się również zegar czasu rzeczywistego, zewnętrzny sterownik stacji dyskietek lub innych urządzeń zewnętrznych, interfejs RS-232 i modemu. Ponadto, mamy kontroler drukarki, wyświetlacza LCD i interfejs klawiatury.

PC-5000 może pracować zarówno podłączony do zasilacza, jak i na wewnętrznym akumulatorze, który zapewniał osiem godzin ciągłej

pracy. Pełne naładowanie trwa około czterech godzin, dlatego producent zalecał zakup dodatkowego akumulatora.

## Drukarka

Drukarka wbudowana w PC-5000 jest jednostką termiczną, która wykorzystuje pojedyncze arkusze papieru. Instalacja drukarki w przypadku, gdy nie jest ona zakupiona z jednostką podstawową, jest prosta. Trzeba tylko podłączyć kabel, oraz wkręcić cztery śruby. W trybie pracy termicznej wystarczy włożyć do drukarki - od tyłu - kawałek papieru termicznego.

Załadunek papieru był znacznie prostszy niż w przypadku podobnych drukarek innych firm, ponadto posiadała ona automatyczny czujnik papieru zatrzymujący drukowanie, gdy skończy się papier. Kolejną ciekawostką to fakt, że na głowicy drukującej znajduje się przełącznik kontrastu, który może być użyty do uzyskania jak najlepszego wydruku.

Drukowanie pojedynczej strony z odstępami zajmuje około dwóch minut, co oczywiście nie jest demonem prędkości, ale również nie jest bardzo wolne. Co więcej, mechanizm termiczny jest naprawdę cichy. Drukować można także na zwykłym papierze, jednak wtedy do drukarki wkładana jest jednorazowa taśma termiczna.

W tym trybie pracy wydrukowany obraz jest "wypalany" na taśmie i przenoszony na papier. Nie rozmazuje się i jest podobny do dobrej kopii typu ksero. Podobnie jak w przypadku kserokopiarki, nie można używać papieru tekstuowanego, a



najlepsze rezultaty daje gładki papier.

## System operacyjny

Standardowym systemem operacyjnym w pamięci ROM komputera jest MS-DOS. Oczywiście system musi być uruchomiony z zainstalowanym napędem dyskowym. MS-DOS w tej wersji posiada wbudowane typowe narzędzia, a dodatkowo specyficzne dla PC-5000 instrukcje zatrzymania i wznowienia przewijania zawartości ekranu, włączenie tak zwanego "echa" drukarki, ustawienia parametrów komunikacyjnych oraz uśpienia urządzenia.

Jeśli chodzi o drukarkę, wprowadzane polecenia i inne teksty mogą być automatycznie kierowane na papier, co pozwala zmienić PC-5000 w elektroniczną maszynę do pisania. Dzisiaj nie myślimy raczej o

takim sposobie pracy, ale 30 lat temu była to jedna z zalet komputera - naśladowanie starszych maszyn.

## Basic

Wbudowany język programowania to GW Basic firmy Microsoft. Zmienne mogą być liczbami całkowitymi, z pojedynczą lub podwójną precyzją albo ciągiem znaków. Domyślnym trybem jest pojedyncza precyzja. Jest to więc typowy dialekt, ale nie brakuje mu też kilku nietypowych rozwiązań. Interesującą instrukcją jest OPTION BASE, która ustawia pierwszą wartość w wielowymiarowej tablicy na 0 lub 1. Tablice mogą mieć do trzech wymiarów.

Możliwości graficzne są szerokie i obejmują polecenia takie jak PSET, RESET i LOCATE (pozyjonowanie kursora), a także CIRCLE do rysowania okręgów. Polecenie WAIT wstrzymuje wykonywanie programu do momentu pojawienia się określonego



bitu na zadanym porcie. Możliwości dźwiękowe są niestety mocno ograniczone, bo komputer posiada bardzo mały wbudowany głośnik, ale w sprzęcie z epoki nie jest to aż tak dziwne.

## Edycja tekstu

Standardowym pakietem do przetwarzania tekstu dostarczanym wraz z komputerem PC-5000 jest SuperWriter firmy Sorcim Software.

Jak na swoje czasy, był dość przyjazny dla użytkownika i nie zawierał rażących błędów zdarzających się w innych programach. Pakiet jest w pełni obsługiwany przez rozwijane menu. Co ciekawe, zawartość ekranu w tym wypadku nie jest przewijana. Zamiast tego, gdy zostanie wpisanych siedem linii, tekst przeskakuje w górę o pięć linii.

Program posiada też wady. Jedną z niezbyt przyjaznych funkcji jest to, że

kursor znika, jeśli jest przesuwany dłużej. W zasadzie, funkcja ta działa prawidłowo, ale trudno jest ocenić, gdzie pojawi się kursor, dlatego trzeba naciskać klawisze pojedynczo, aby nie stracić orientacji. Tryb wstawiania jest dostępny, ale kursor nie zmienia rozmiaru ani kształtu, więc trudno sprawdzić, czy jesteśmy w trybie wstawiania. Po prostu trzeba obserwować ekran podczas pisania tekstu.

Wraz z komputerem PC-5000 dostarczany był też pakiet oprogramowania biurowego, mianowicie - SuperCalc-2 i SuperPlanner. Był więc od razu dostępny rozbudowany arkusz kalkulacyjny. Ponadto producent miał podpisane umowy z ok. 20 firmami pracującymi nad oprogramowaniem dla nowego sprzętu.

## Podsumowanie

Jak ocenić produkt firmy Sharp? Wiele osób traktuje go dzisiaj jak starego laptopa, ale założenie twórców było nieco inne. Potraktowano MS-DOS jako platformę do osiągnięcia popularności w USA, przyjmując jako główne zastosowanie edycję tekstu - stąd wbudowana drukarka. Podobnych konstrukcji było więcej, na przykład firmy Canon, ale nie przyjęły się one w dłuższej perspektywie.

Z pewnością powodem były zbyt duże ograniczenia rozbudowy, jak również dynamiczny rozwój technologii stosowanych w drukarkach. Trzeba jednak przyznać, że sam pomysł stworzenia maszyny typu "All-In-One" był bardzo dobry i był to odważny krok japońskiego producenta.

**SHARP INTRODUCES**  
**COMPUTERS TO GO**

With Sharp's PC-5000 you can take your entire office with you, wherever you go. Do word processing on the train, order entry from a customer's office or spread sheet analysis in your hotel room. It goes anywhere.

- It's small.
- It weighs under 10 lbs\* and fits in a standard briefcase.
- It prints. Quietly, with an optional correspondence-quality printer.
- It comes with software. Word processing and communications. Also available are spread sheeting, executive planning and scheduling.
- It remembers over 80 (128K) typewritten pages. And can be expanded to handle over 500 (896K).
- It's compatible with a wide array of 16-bit IBM® software.
- It communicates with other computers and databases.
- It's AC/DC with rechargeable 6-hour batteries.
- It all means travel time and commuter time no longer have to be downtimes.

For more information call toll-free now 1-800-BE SHARP or send in the coupon.

**SHARP**  
FROM SHARP MINDS COME SHARP PRODUCTS

Sharp Electronics Corp., 1900 E. Cornell, Peoria, IL 61614  
☐ Please send me more information about Sharp's PC-5000.  
☐ Please set up a demonstration.

Name \_\_\_\_\_ Title \_\_\_\_\_  
Company \_\_\_\_\_  
Street \_\_\_\_\_ City \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_  
Phone (\_\_\_\_) \_\_\_\_\_  
Dept. S-8-5-7-4

\*Without optional printer and modem.  
IBM® is a registered trademark of International Business Machines Corporation.  
Sharp Electronics Corp., Paramus, NJ 07652

# Sinclair QL

## Trochę inny komputer

**MARCIN LIBICKI**

**Dla firmy Sinclair Research wprowadzenie komputera QL w 1984 roku było odejściem od domowych komputerów serii ZX Spectrum klasy niskiej i pierwszą próbą wejścia na rynek biznesowy. Sinclair QL to komputer naprawdę wyjątkowy, choć nigdy nie zdobył większej popularności.**

Maszyna posiada 32-bitowy procesor Motorola 68008 oraz 128 kilobajtów pamięci RAM. Zamiast magnefonu lub stacji dyskietek, QL ma wbudowane dwa napędy Microdrive, o których więcej napiszę w dalszej części artykułu. Klawiatura ma 65 klawiszy, w tym pięć klawiszy funkcyjnych po lewej stronie. QL szczyci się również wielozadaniowością, możliwością uruchomienia systemu okienkowego i pakietem oprogramowania napisanym przez firmę Psion.

Cały komputer jest mniej więcej takiej samej wielkości jak odłączane klawiatury komputerów stacjonarnych. Wcześniejsze prototypy posiadały cartridge podłączany z tyłu, jednak z tego co wiem, pomysł został zarzucony, bo pojawiła się obawa, czy wszystko zmieści się w małej obudowie w wersji produkcyjnej.

Komputer QL korzysta z procesora Motorola 68008 pracującego z częstotliwością 7,5 MHz, co sprawia, że

jest dość szybki w porównaniu do ówczesnych odpowiedników. Maszyna wykorzystuje wiele niesztywnych układów scalonych dla obsługi operacji wejścia-wyjścia, grafiki i systemu operacyjnego.

Mamy tutaj trzy 16-kilobajtowe układy ROM zawierające system operacyjny QDOS oraz wersję Basic firmy Sinclair, zwaną teraz SuperBasic. QL posiada także 128 kilobajtów pamięci RAM, która nie była przewidziana do rozbudowy wewnętrznej. Producent obiecywał jednak możliwość dodania 512 kilobajtów RAM za pomocą zewnętrznego rozszerzenia pamięci.

QL posiada dwa porty szeregowo RS-232, dwa porty o nazwie "QL Net" przeznaczone do łączenia z innymi komputerami QL oraz dwa porty joysticka. Nie ma portu równoległego dla drukarki, bo podobnie jak w przypadku Apple IIc, Sinclair wierzył, że większość użytkowników będzie podłączała drukarkę do portu szeregowego. Można powiedzieć, że

słynny wynalazca wyprzedził w tym wypadku swoją epokę, bo dzisiaj przecież każdy podłącza drukarkę do portu USB, który de facto jest nowoczesną wersją portu Serial.

Oryginalny zasilacz QL nie ma włącznika, jednak po prawej stronie obudowy komputera mamy przycisk Reset. Klawisze są bardzo charakterystyczne, ponieważ są kwadratowe, ale każdy ma zaokrąglone wgłębienie w górnej części. Do takiej klawiatury trzeba się przyzwyczaić.

W przeciwieństwie do poprzednich komputerów firmy Sinclair, QL nie umożliwia wprowadzania słów kluczowych przypisanych do określonych klawiszy. Klawiatura wygląda typowo, posiada klawisze ESC, CTRL, ALT oraz funkcyjne od F1 do F5. Pierwsze trzy klawisze funkcyjne są zdefiniowane w ten sposób, że F1 wywołuje podręczną pomoc, F2 włącza i wyłącza górny obszar podpowiedzi na ekranie, a F3 uruchamia menu poleceń.

Niestety, klawisze kierunkowe kursora znajdują się po obu stronach Spacji. Strzałki w lewo i w prawo w połączeniu z klawiszem CTRL działają jako klawisze kasowania.

Oprogramowanie jest dość rozbudowane, dlatego używając różnych kombinacji klawiszy SHIFT, ALT i CTRL z klawiszami kursora, można usuwać całe słowa lub linie i szybko przemieszczać kursor po edytowanym tekście. Podobnie jak wiele komputerów z niższej półki, QL nie miał wskaźnika aktywności klawisza CAPS LOCK.

Napędy Microdrive w momencie premiery komputera QL były już sprzedawane od około roku, a Sinclair włożył dużo wysiłku rozwój tych urządzeń. W rzeczywistości wewnątrz jest taśma magnetyczna, ale wczytywanie programów z takiego napędu jest bardzo szybkie.

Microdrive dla QL były inaczej formatowane niż dla ZX Spectrum, ale mechanizm pozostał taki sam. Każda taśma ma teoretyczną pojemność 128 kilobajtów, choć nie cały obszar jest dostępny dla użytkownika. Formatowanie można za to wykonać bardzo łatwo, za pomocą jednego z poleceń wbudowanego języka Basic.

Gdy komputer QL jest włączony lub wciśnięty zostanie przycisk resetowania, na ekranie pojawi się komunikat:

**F1 for monitor;**  
**F2 for television.**

Po naciśnięciu przycisku F2 wyświetlany jest 64-kolumnowy ekran. Szkoda, że nie ma tutaj pełnego trybu 80-kolumnowego, ponadto jeśli chcemy korzystać ze starszych

programów będzie trzeba uruchomić tryb 40-kolumnowy. Jest on nieco "oszukany", bo linia mieści tylko 37 znaków.

Do maszyny można podłączyć monitor RGB lub Composite Video. W trybie monitora, ekran o szerokości 80 znaków jest automatycznie dzielony na dwa pionowe okna. Lewe okno pokazuje listę programów, a wszystkie operacje są uruchamiane w prawym oknie. Domyślnie okna mają połowę szerokości ekranu, można jednak określić własny rozmiar.

Wewnątrz okna można przewijać zawartość w pionie poleceniem SCROLL i w poziomie poleceniem PAN. Co ciekawe, polecenia te działają w dwóch kierunkach (w górę i w dół oraz w prawo i w lewo). Niestety instrukcje te nie działają do końca dobrze na oryginalnym oprogramowaniu, bo po przewinięciu informacje są czyszczone z okna i nie można ich odzyskać bez ponownego wyświetlenia przez program.

Tekst może być wyświetlany w trybie 40, 64, lub 80 znaków w poziomie oraz 25 linii. Rozdzielczość graficzna to 512x256 pikseli w czterech kolorach lub 256x256 w ośmiu kolorach.

System graficzny zaimplementowany w SuperBasic jest dość wydajny i automatycznie "poprawia" kształt pikseli podczas rysowania kół i innych figur geometrycznych. Polecenie SCALE pozwala na zmianę skali rysunku oraz przesunięcie punktu początkowego. Inne polecenia graficzne to CIRCLE, ARC, LINE, POINT i FILL.

Ogólnie mówiąc, SuperBasic jest bardzo zbliżony do Microsoft Basic,

ale posiada kilka dodatkowych instrukcji graficznych obsługujących okna oraz język Logo. Oprócz komend graficznych, QL posiada także polecenie MODE, która umożliwia przełączanie rozdzielczości poziomą ekranu między trybem 512 i 256 pikseli.

Aby ułatwić edycję programu, SuperBasic posiada automatyczną numerację linii. Jeśli chodzi o polecenia dla Logo, są to instrukcje PENUP, PENDOWN, MOVE, TURN, i TURNT0, za pomocą którego nasz żółw zmienia kierunek. Większość pozostałych poleceń jest bardzo podobna do Microsoft Basic, ale składnia niektórych z nich jest zmodyfikowana.

Niektóre instrukcje wymagają stosowanie odstępów między argumentami, inne można wpisywać bez znaków Spacji, co nie jest to zbyt wygodne. Procedury są definiowane z argumentami podawanymi w nawiasach, ale należy je wywoływać już bez nawiasów.

SuperBasic i dołączony oryginalnie pakiet oprogramowania firmy Psion, działają w systemie QDOS. Jest to środowisko wielozadaniowe, dzięki czemu do pliku może uzyskać dostęp więcej niż jeden proces jednocześnie. Programy mogą dzielić plik lub uzyskać wyłączny dostęp z możliwością określania praw dostępu do odczytywania lub zapisywania informacji.

Z pewnością w oprogramowaniu komputera QL można znaleźć bardzo wiele ciekawych rozwiązań, niestety w naszym magazynie nie mamy aż tyle miejsca. Dlatego tym tematem będę się zajmować dalej, w następnych numerach RetroKompa.

# Sinclair Basic

## Zastosowanie funkcji

Opracował:  
**MARIUSZ  
WASILEWSKI**

W programowaniu mianem funkcji określa się fragment kodu, który może być wykonywany wielokrotnie z różnych miejsc programu. Mówi się, że funkcje "zwracają" wynik, który jest zapisywany następnie w zmiennej. W Basicu można korzystać z funkcji matematycznych lub takich, które nie są związane z liczbami. Mogą one podawać zupełnie innego rodzaju informacje, a stopień skomplikowania zależy głównie od pamięci w komputerze.

W Sinclair Basicu możesz również samodzielnie definiować funkcje. Dozwolona nazwa dla takiej funkcji składa się z FN i litery (jeśli wynik to liczba) lub FN, litery i znaku \$ (jeśli wynik jest ciągiem tekstowym). Funkcje te są bardziej restrykcyjne na temat nawiasów: argument musi być umieszczony w nawiasach.

Funkcję definiujemy, umieszczając w programie polecenie DEF. Na przykład, oto definicja funkcji FN s, której wynikiem jest kwadrat argumentu:

```
10 DEF FN s(x)=x*x:
REM kwadrat x
```

DEF uzyskuje się w trybie rozszerzonym, używając SYMBOL SHIFT i 1. Gdy wpisujemy to, komputer pokaże automatycznie FN, ponieważ w poleceniu DEF za słowem DEF zawsze występuje bezpośrednio FN. Na koniec s zamyka nazwę funkcji FN s.

Zmienna x w nawiasach jest nazwą, za pomocą której możemy się odwoływać do argumentu funkcji. Do tego zadania można użyć dowolnej jednoliterowej nazwy lub w przypadku argumentu tekstowego — nazwy jednoliterowej ze znakiem \$.

Po znaku = następuje właściwa definicja funkcji. Może to być dowolne wyrażenie i może ono odwoływać się również do argumentu, wykorzystując nadaną mu nazwę. W naszym przypadku jest to x. Reasumując, działamy tak samo, jakby była to zwykła zmienna.

Gdy wprowadzimy ten wiersz, możemy wywołać zdefiniowaną funkcję w taki sam sposób, jak funkcje własne komputera, wpisując jej nazwę FN s, a następnie argument. Należy pamiętać, że dla funkcji zdefiniowanych samodzielnie, argument musi być umieszczony w nawiasach. Na przykład:

```
PRINT FN s(2)
PRINT FN s(3+4)
PRINT 1+INT FN s(LEN "kur-
czaki"/2+3)
```

Po umieszczeniu w programie odpowiedniego polecenia DEF, możemy używać własnych funkcji w wyrażeniach tak samo swobodnie, jak funkcji komputera.

### Uwaga!

W niektórych dialektach języka BASIC należy w nawiasach umieszczać nawet argumenty funkcji własnych komputera. Tak jednak nie jest w Basicu na ZX Spectrum.

Funkcja INT zawsze zaokrągla w dół. Aby zaokrąglić do najbliższej liczby całkowitej, dodajemy najpierw .5. Możemy też napisać własną funkcję, która robi to samo:

```
20 DEF FN r(x)=INT
(x+.5):
```

```
REM daje x zaokrąglone do
najbliższej liczby
całkowitej.
```

Wtedy otrzymamy przykładowo:

```
FN r(2.9) = 3
FN r(2.4) = 2
FN r(-2.9) = -3
FN r(-2.4) = -2
```

Porównajmy te wyniki z sytuacją, gdy zamiast FN r zastosujemy samo INT.

Oto kolejny program:

```
10 LET x=0: LET y=0:
   LET a=10
20 DEF FN p(x,y)=a+x*y
30 DEF FN q()=a+x*y
40 PRINT FN p(2,3),FN q()
```

Występuje w nim jednak wiele ciekawych elementów. Po pierwsze, funkcja nie jest ograniczona do jednego argumentu: może posiadać ich więcej lub nawet żadnego — lecz wciąż muszą pozostać wpisane nawiasy.

Po drugie, nie ma znaczenia, gdzie w programie umieścimy polecenia DEF. Gdy komputer wykona wiersz 10, po prostu przeskoczy linie 20 i 30, aby przejść do wiersza 40. Jednak muszą one być umieszczone w programie. Nie można zapisać ich w trybie bezpośrednim, czyli bez numeru wiersza.

Po trzecie, x i y są zarówno nazwami zmiennych w programie jako całości, jak i nazwami argumentów funkcji FN p. Funkcja ta tymczasowo "zapomina" o zmiennych x i y, lecz nie mając na liście argumentów a, wciąż "pamięta" o zmiennej a. Stąd, gdy FN p(2,3) jest obliczane, posiada



wartość 10. Zmienna, x ma wartość 2, ponieważ jest to pierwszy argument, a y ma wartość 3, ponieważ jest drugim argumentem. Wynikiem jest więc  $10+2 \times 3=16$ .

Z drugiej strony, gdy FN q() jest obliczane, nie ma żadnych argumentów, zatem a, x i y wciąż odnoszą się do zmiennych i posiadają wartość odpowiednio 10, 0 i 0. Wynikiem w tym przypadku jest  $10+0 \times 0=10$ .

Gdy linię 20 zmienimy na następującą:

```
20 DEF FN p(x,y)=FN q()
```

tym razem FN p(2,3) da wynik 10, ponieważ FN q wciąż wraca do zmiennych x i y, a nie do argumentów funkcji FN p.

Niektóre języki Basic posiadają też funkcje o nazwach LEFT\$, RIGHT\$, MID\$ i TL\$. Z pewnością warto je poznać:

**LEFT\$(a\$,n)** - daje ciąg tekstowy a\$ składający się z n pierwszych znaków.

**RIGHT\$(a\$,n)** - daje ciąg a\$ składający się z n ostatnich znaków.

**MID\$(a\$,n1,n2)** - daje ciąg a\$ składający się z n2 znaków zaczynając od pozycji n1.

**TL\$(a\$)** - daje ciąg a\$ zawierający wszystkie znaki z wyjątkiem pierwszego.

Możemy napisać kilka funkcji, które będą wykonywały to samo, na przykład:

```
10 DEF FN t$(a$)=a$(2 TO
   ): REM TL$
20 DEF FN l$(a$,n)=a$( TO
   n): REM LEFT$
```

Teraz trzeba sprawdzić, czy funkcje te dobrze działają z ciągami tekstowymi o długości 0 lub 1.

Nasza funkcja FN l\$ ma dwa argumenty - jeden jest ciągiem, a drugi jest liczbowy. Funkcja może posiadać do 26 argumentów liczbowych, a w tym samym czasie również do 26 argumentów traktowanych jako ciągi, czyli zawierające tekst.



# Logo

## Wypełnianie obszarów

Opracował:  
**MARIUSZ  
WASILEWSKI**

Logo to prawdziwy język-legenda. Być rozpowszechniony w latach '80-tych, gdy w naszym kraju pojawiały się pierwsze pracownie komputerowe. Zazwyczaj nauczyciele pokazywali podstawowe instrukcje tak, aby zachęcić młodych ludzi to spojrzenia w innym kierunku niż machanie joystickiem. Jednak w stosunkowo prosty sposób można także tworzyć bardziej skomplikowane algorytmy.

Język Logo nie posiada polecenia wypełniającego obszary na ekranie. Mogłoby to wyglądać tak, że jeśli umieścisz żółwia wewnątrz jakiegoś kształtu i wywołasz instrukcję, kształt wypełni się kolorem. W Logo można tworzyć wszelkiego rodzaju proste i skomplikowane projekty graficzne, więc wypełnienie pozwoliłoby w prosty sposób rysować na przykład szachownice.

Gdy już wiemy, jakie są nasze potrzeby możemy napisać prosty program, który rzeczywiście zrealizuje taką funkcję. Zaczniemy od najprostszego kształtu, czyli kwadratu

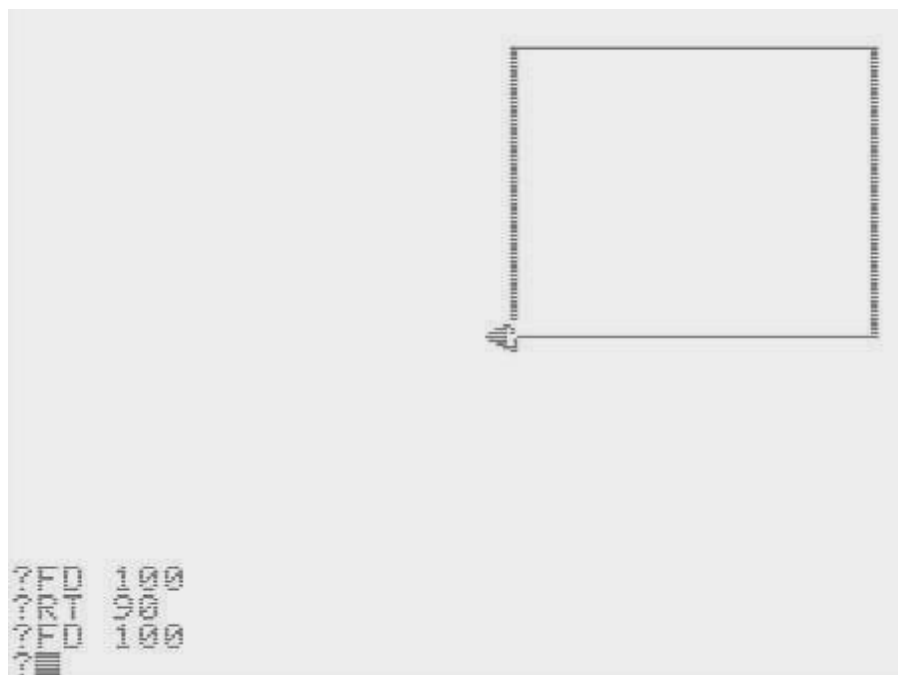
### Kolorowanie kwadratu

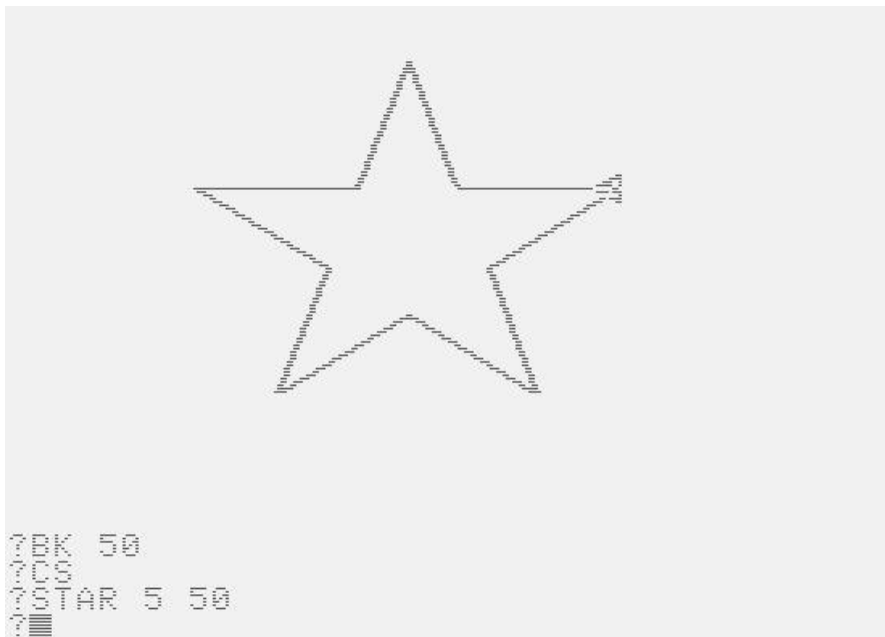
Na początek proponuję narysować kwadrat na kartce papieru i pomalować go na jeden kolor. Od razu można zauważyć, że jednym z naturalnych sposobów kolorowania jest poruszanie się w przód i w tył, przesuwając się w dół figury w miarę jak

się poruszamy. Jak można powiedzieć to samo naszemu żółwiowi? Jednym ze sposobów jest powtarzanie następujących poleceń tyle razy, ile jest to konieczne:

```
FD :D RT 90 FD 1 RT 90
FD :D LT 90 FD 1 LT 90
```

Jak przy tworzeniu większości algorytmów w języku programowania takim jak Logo, istnieje więcej niż jedno rozwiązanie naszego problemu. Używając rekurencji i warunków możemy napisać procedurę rysowania kolejno mniejszych kwadratów, aż do całkowitego





wypełnienia kształtu, przykładowo tak:

```
TO FILL.SQUARE :N
IF :N = 0 [STOP]
REPEAT 4 [FD :N RT 90]
FILL.SQUARE :N - 1
END
```

Dalej, skoro możemy to zrobić z kwadratem, możemy powtórzyć to z inną figurą, na przykład z trójkątem:

```
TO FILL.TRIANGLE :N
IF :N = 0 [STOP]
REPEAT 3 [FD :N RT 120]
FILL.TRIANGLE :N - 1
END
```

Jeśli ten sam algorytm działa w stosunku do trójkąta, powinniśmy być w stanie to zrobić dla sześciokąta, a tak naprawdę - z każdym zwykłym wielokątem. Dlatego teraz się tym zajmiemy.

## Rysowanie wielokątów

Jeśli kiedykolwiek rysowałeś wielokąty to jest jasne, że żółw musi

obrócić się w tym wypadku o 360 stopni. Zostanie wtedy narysowany wielokąt, a dodatkowo żółw zakończy swoją pracę ustawiony w tym samym kierunku, w którym zaczął rysować. To ważne, bo wiedząc o tym prostym fakcie możemy napisać procedurę, która narysuje dowolny wielokąt, jaki chcemy uzyskać:

```
TO POLY :R :D
REPEAT :R [FD :D RT
360/:R]
END
```

Skoro tak, powinniśmy być w stanie napisać kolejną procedurę, która wypełni naszą figurę:

```
TO FILL.POLY :R :D
IF :D = 0 [STOP]
REPEAT :R [FD :D RT
360/:R]
FILL.POLY :R :D - 1
END
```

Dzięki temu można szybko odkryć, że procedura działa całkiem dobrze dla trójkątów, kwadratów, pięciokątów i sześciokątów. W przy-

padku wielokąta z więcej niż sześcioma bokami będziemy mieli luki w wypełnieniu. Im więcej ścian figury, tym więcej uzyskamy niewypełnionych części. Może to wyglądać interesująco, ale jeśli jesteś bardziej zainteresowany wypełnianiem kształtów, musimy zmodyfikować procedurę, aby działała lepiej.

Zrobiłem wiele prób i gotowy algorytm wygląda teraz tak:

```
TO FILL.POLY2 :R :D
IF :D = 0 [STOP]
REPEAT :R [FD :D RT
360/:R]
RT 90 FD 1 LT 90
FILL.POLY2 :R :D - .5
END
```

## Rysowanie okręgów

Oczywiście znowu można napisać cały program w różny sposób, ale moja wersja działa prawidłowo w stosunku do wielokątów od 7 do 14 boków. Gdy uzyskasz 14 boków figura będzie wyglądała jak koło. Wielokąt o 15 ścianach zaczyna znowu mieć kilka luk, dlatego trzeba ponownie edytować procedurę.

Pokazane procedury są wystarczające do zabawy w języku Logo na wielu różnych komputerach. Na Amstradzie możemy uruchomić Dr Logo i pisać całkiem rozbudowane programy. Przypominam, że jeśli nie wiesz co znaczą różne instrukcje można użyć słowa HELP wpisanego osobno lub z nazwą polecenia.

Logo może być świetną propozycją do nauki prostego programowania dla naszych dzieci i warto też o tym pamiętać.

# Rodzina Commodore 264

**MICHAŁ „stRing”  
RADECKI-MIKULICZ**

**Od redakcji: W naszym magazynie komputery takie jak C16, C116 i Plus/4 były już opisywane wielokrotnie. Jednak tym razem Autor nowego artykułu przedstawia bardziej praktyczne podejście do tego sprzętu. Zachęcamy do lektury!**

## Wstęp i odrobina historii

Wystarczy spojrzeć na produkcje z kilku ostatnich lat (Pets Rescue, Rocket Science, Metamerism), żeby stwierdzić, że rodzina 264 (\*) miała duży potencjał, mimo iż w założeniach miały to być tanie i proste komputery. No może pomijając Plusa 4, który miał być przeznaczony do bardziej zaawansowanych zadań, w tym przeróżnych prac biurowych (edytory tekstów, arkusze kalkulacyjne itd.).

Plus 4 miał dzięki temu dwie kości ROM z dołączonym oprogramowaniem, dodatkowy port rozszerzeń i 64KB pamięci RAM na pokładzie. Pozostałe modele (C16 i budżetowy C116 z gumową klawiaturą podobną jak w ZX Spectrum) posiadały jedynie 16KB pamięci i były pozbawione wspomnianych rozszerzeń. Poza tym we wszystkich modelach znajdziemy te same układy scalone (procesor 7501/8501, wielozadaniowy układ TED, Basic w ulepszonej, w stosunku

do Commodore 64, wersji 3.5 – dodano m.in. komendy do obsługi stacji dysków, czy monitor z możliwością wpisywania komend w assemblerze; oraz PLA i Kernal). Wszystko to na 1984 rok, kiedy modele zostały wypuszczone na rynek, wydawało się całkiem przyzwoitą platformą i jedynie brak układu dźwiękowego z prawdziwego zdarzenia mógł być piętą achillesową tychże komputerów. Wówczas nawet brak sprzętowych duszków, których nie posiadają opisywane modele, dałoby się przeboleć.

Patrząc obiektywnie na lata 80. i na to co firma Commodore osiągnęła wcześniej, jednocześnie analizując kierunek w jakim szła, był to jednak mały krok wstecz. Po obiecującym PET, dobrym VIC'u i wyśmienitym C64, publika miała prawo oczekiwać czegoś lepszego. Nawet montując do serii 264 SID'a w jakiejś wzbogaconej wersji (więcej kanałów, stereo) byłby to krok w dobrą stronę, ponieważ w mojej ocenie, właśnie brak dźwięku z prawdziwego zdarzenia jest najmocniej odczuwal-

ny. Pozostaje jeszcze problem pozostałych, wybrakowanych pod względem ilości pamięci, członków rodziny 264. Rezygnując z modelu C16, lub nawet dodając mu więcej pamięci i eliminując tym sposobem możliwość tworzenia bazy programowej, która ze względów ekonomicznych byłaby pisana głównie pod 16KB (aby dało się ją uruchomić na obu sprzętach), można by spowodować, iż wszystko byłoby tworzone bez ograniczeń - pod 64KB i po prostu lepiej wyglądać i funkcjonować. Po co tworzyć taką linię komputerów, gdzie każdy z nich wyklucza się wzajemnie, w sensie jednej bazy programowej, a wszystko pisane będzie głównie pod 16KB?

Tak, Plus 4 miał de facto nie być komputerem do gier, ale nie spełnił się także i na tej materii – a do tego przecież był zaprojektowany. Wystarczy uruchomić jego wbudowane oprogramowanie biurowe, żeby się o tym przekonać. Jest mało intuicyjne, ubogie graficznie i obsługiwane wyłącznie przy pomocy

klawiatury. Czy patrząc z perspektywy czasu dla firmy Commodore nie lepszym pomysłem od strony marketingowej byłoby wyprodukowanie sprzętu biurowego na wzór Plusa 4 (z dodatkowymi pamięciami ROM jak w oryginalnych założeniach, oprogramowaniem biurowym z lepszą obsługą, być może bez układu dźwiękowego, ale za to z większą pamięcią RAM - może nawet rozszerzalną do 256KB), oraz równoległe sprzętu typowo „gaming'owego” opartego na podobnej, ale tańszej konstrukcji, dodatkowo z ulepszonym SID'em? Takiego 8-bitowego następcy C64 z prawdziwego zdarzenia? To tylko luźne dywagacje, zatem skupmy się na konstrukcji istniejących modeli.

## Wady i zalety

We wszystkich modelach wprowadzono przycisk reset - doskonałe rozwiązanie, które ułatwia pracę. Kolejną z zalet C16 jest jego klawiatura - sprawdzony, doskonały od strony mechanicznej i niezawodny system (potwierdziło się to już przy VIC'u i C64). Miałem do czynienia z wieloma komputerami retro i klawi-

iatury od VIC, C64 i C16 po dziś dzień wyglądają solidnie, są wygodne i z reguły nie wymagają napraw. W C16 poprawki wymagałyby jednak kursory, bo moim zdaniem nawet w C64 jest to lepiej zaprojektowane, mimo iż także wymaga wprawy. Klawiatura Plusa 4 jest też znośna, ale nie najlepsza - doceniam fakt umieszczenia kursorów z boku, mimo że te klawisze mogłyby być większe.

Z kolei gumowa klawiatura w C116 jest niezbyt wygodna. Nie nazwałbym jednak tego katastrofą, ponieważ model ten z założenia miał konkurować z prostymi modelami od Sinclair'a i miał być po prostu tani. W założeniach i planach Tramiela, cena sprzedażowa tego egzemplarza miała wynosić około \$50 (Plus 4 miał kosztować drugie tyle), jednakże finalnie komputery, po odejściu Tramiela z firmy były sprzedawane znacznie drożej.

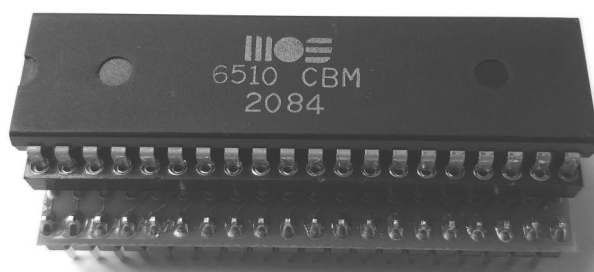
Rodzina 264 jest dość podatna na uszkodzenia. Jeśli zatem traficie na działający egzemplarz - zadbajcie o niego. Procesory 8501/7501 i układ TED 8360 (Text EDitor, lub Text Editing Device - w zależności od źródła)

dość często okazują się niesprawne. Układy te są bardzo podatne na uszkodzenia elektroniczne i przegrzanie. Czterdziestoosmio nóżkowy TED jest ciekawym i nowatorskim jak na ówczesne czasy układem wielozadaniowym.

Oprócz generowania grafiki i dźwięku, jego zadaniem jest między innymi także odświeżanie pamięci RAM, czy obsługa danych z klawiatury. Oba czipy były stosowane wyłącznie w modelach serii 264, więc znalezienie identycznych zamienników z innych komputerów i urządzeń nie jest możliwe. Jeśli jesteście w posiadaniu innego komputera z serii 264 w którym sprawny jest chociaż jeden z wymienionych układów to mamy szansę odratować nasz egzemplarz. Co jednak z tego, skoro drugi komputer pozostanie wybrakowany?

## Na procesor jest metoda

Obecnie możemy przynajmniej poradzić sobie z wadliwym procesorem na kilka sposobów. Pierwszy to zaopatrzyć się w specjalny adapter, który umożliwi nam wykorzystanie procesora 6510 lub 8500 (montowanego w C64 i dużo łatwiej dostępnego i w przystępnej cenie). Można też pokusić się o stworzenie takiej płytki samemu, ponieważ adapter jest dość prosty wykonaniu - nie posiada żadnych elementów elektronicznych, jedynie przekierowuje sygnały do innych pinów na procesorze. Wadą zastosowania zamiennika procesora jest brak możliwości zatrzymania przez komputer silnika w magnetofonie, który będzie działał bez przerwy (co stanowi raczej drugorzędny problem, jeśli nie



6510 CPU i adapter.

korzystamy z tego urządzenia), oraz co istotniejsze - brak możliwości wykorzystania szybkich „loaderów” podczas ładowania danych ze stacji dysków.

Oznacza to że nie uruchomimy pomniejszej części oprogramowania, w tym kilku nowszych gier (zapisanych w obrazach dysków D64), czy niektórych produkcji demoscenowych. Poza tym komputer będzie działał prawidłowo. Do stuprocentowej funkcjonalności (bo nie mamy jeszcze dostępu do stacji dysków czy jej emulatora sprzętowego) będziemy musieli podmienić oryginalny kernal, w którym zapisane są na nowo procedury dotyczące obsługi portów. 6510/8500 ma sześć portów wejścia/wyjścia P0-P5, a 8501/7501 ma ich siedem P0-P6. Cały komplet - adapter, procesor i kernal możemy zakupić na eBay'u (sprzedawca hackjunkcom) w cenie 30 funtów plus wysyłka. Na stronie:

<http://hackjunk.com/2017/06/23/commodore-16-plus-4-8501-to-6510-cpu-conversion/>

znajdziemy też więcej szczegółów odnośnie adaptera i ewentualnej (choć nie wymaganej) przeróbki na płycie głównej komputera. Konieczne będzie wlutowanie jednej diody prostowniczej, gdy jednak zechcemy korzystać z magnetofonu. Umożliwi nam to normalną pracę z tym urządzeniem - będzie on uruchamiał silnik we właściwych momentach, czyli podczas przewijania i po wciśnięciu play. Przed zakupem adaptera, zwróćmy uwagę na model komputera w którym będziemy go instalować - do wyboru mamy krótszą wersję płytki - pasuje do modelu

C16 i dłuższą - do zamontowania w Plus 4. Z tego co udało mi się ustalić, wersja krótka zmieści się także w modelu C116, ale tu będziemy zmuszeni wlutować adapter na stałe, ponieważ jego wysokość uniemożliwi nam całkowite zamknięcie górnej części obudowy z klawiaturą. W chwili gdy będziemy chcieli zamienić adapter na oryginalny procesor 8501/7501 konieczne będzie jednak ponowne rozlutowanie zestawu.

Drugą opcją na zastąpienie wadliwego procesora jest wstawienie w jego miejsce płytki opartej na FPGA emulującej 8501/7501. Jest to doskonałe rozwiązanie i stosunkowo nie drogie w porównaniu do poprzedniej metody. Koszt takiego urządzenia to około 250 PLN. Można je co jakiś czas zakupić u użytkownika Zaxon na sellmyretro.com. Przy zastosowaniu tej sprzętowej emulacji, nie musimy nawet zmieniać kernala, bo mamy pełną emulację oryginalnego procesora. Jedyne problemy jakie napotkałem podczas użytkowania tego modułu, to nie do końca poprawna współpraca z SD2IEC - od czasu do czasu występują sporadyczne przerwy podczas wczytywania danych i należy zresetować komputer. Być może wina leży po stronie samego SD2IEC, oprogramowania, lub samego komputera, jednakże przy korzystaniu z Ultimate II+(\*\*) nie zaobserwowałem podobnych problemów. Bądź co bądź, podczas testów, udało mi się wczytać z SD2IEC większość produkcji.

Oczywiście możemy szukać oryginalnego procesora 8501/7501 na podmiętę, jednakże znalezienie takiego graniczy z cudem i wymaga posiada-

nia zasobnego portfela. Poszukiwania prowadzimy głównie na eBay i Aliexpress.

## TED i jego bolączki

Gorzej, jeśli naszym pacjentem okaże się TED, który potrafi polec na różne sposoby i z różnych przyczyn. Póki co nie ma zamiennika tego układu, ale od czasu do czasu można znaleźć używane czipy na eBay'u. Ceny za niego dochodzą nawet do 200 zł, a nie mamy żadnej pewności ile taki pożyje. Jest projekt FPGATed, który podobnie jak zamiennik procesora od Zaxona emuluje układ TED, jest on jednak póki co w fazie projektowania i testów i nie jest dostępny w sprzedaży. Wiele posiadaczy zepsutych modeli serii 264 niecierpliwie czeka na jego ukończenie.

## Jak zadbać o swój komputer?

Zadbanie o chłodzenie oryginalnego procesora 7501/8501 i układu TED jest kluczowe i jak najbardziej wskazane. Ze względów oczywistych nie stosujemy go na procesorze 6510 z adapterem (łączna wysokość procesora, adaptera i chłodzenia byłaby zbyt duża), poza tym po uszkodzeniu, ewentualna dostępność tych procesorów jest póki co bezproblemowa. Radiatory które możemy zastosować są różne. Ja zamocowałem na TED cztery aluminiowe, o wymiarze około 2cm x 2cm z klejem termo-przewodzącym, dedykowane na pamięci nowoczesnych kart graficznych. Podczas pracy radiatory na układzie TED nagrzewały się do temperatury czterdziestu-kilku stopni, więc raczej spełniają swoje zadanie.

Warto wspomnieć, że w Plus 4, jak i C116 układ TED ma chłodzenie fabryczne w postaci miedzianej sprężynującej blaszki, która zamocowania jest do metalowej obudowy ekranującej, która dodatkowo rozprowadza ciepło z nagrzewającego się układu. Radiatory możemy także przykleić do układu PLA (251641), gdyż podczas pracy ten grzeje się dość mocno. W razie awarii PLA nie mamy też większych powodów do paniki, ponieważ obecnie na rynku dostępne są zamienniki tego układu, m.in. SuperPLA, który z powodzeniem może być także użyty w modelach C64. Ustawia się jedynie zworki na zamienniku, w zależności od tego z jakim komputerem ma on pracować.

## Trochę o peryferiach

Do serii 264 przewidziano dedykowane i niestandardowe joysticki. Na płytach tych komputerów są dwa gniazda minidin-8, do których je podłączamy. Zrezygnowano tu ze standardowego złącza DB9 znanego z Atari 8-bit i Commodore (C64, VIC-20). Ich dostępność jest utrudniona, ale istnieje możliwość wykonania lub zakupu adaptera, dzięki któremu podłączymy typowe joysticki do serii 264. Na rysunku przedstawiono sposób połączenia pinów, nie ma w środku żadnej elektroniki – jedynie przesyłane są sygnały. Istotne jest aby SELECT z gniazda minidin-8 podłączyć do GND w gnieździe DB9. Gotowy adapter kupimy np. na eBay za około 50 PLN. Dla wprawionych lutowniczych, złożenie takiej przejściówki nie będzie stanowiło problemu. Razem z serią 264 firma Commodore wypuściła na rynek nową stację dysków, dedykowaną do



Program diagnostyczny Diag264.

tych komputerów. Warto tu nadmienić że do C16/C116 i Plus 4 podłączymy także oryginalną stację od C64, bo urządzenie jest kompatybilne – posiada taki sam port serial. Nowa stacja o oznaczeniu 1551 łączy się jednak z komputerem przez port kartridża, co powoduje dużo szybszą (nawet kilkakrotnie) transmisję danych, niż przez szeregowy port IEC.

Oczywiście jak wspomniano, można do serii 264 podłączyć także nowsze urządzenia, jak SD2IEC, czy Ultimate II+ (jednak przy współudziale C64). Standardowo odczytamy pliki PRG, jak i obrazy dyskietek np. D64. Podłączenie C16/C116 i Plus'a 4 do telewizora lub monitora także nie stanowi problemu jeśli jesteśmy w posiadaniu takiego kabla do C64, czy 8-bitowego Atari. Rodzina 264 posiada wyjście video, oraz chrominację i luminancję jako osobne sygnały, dokładnie jak tamte komputery i na tych samych pinach. DO gniazda video pasuje wtyk DIN-8.

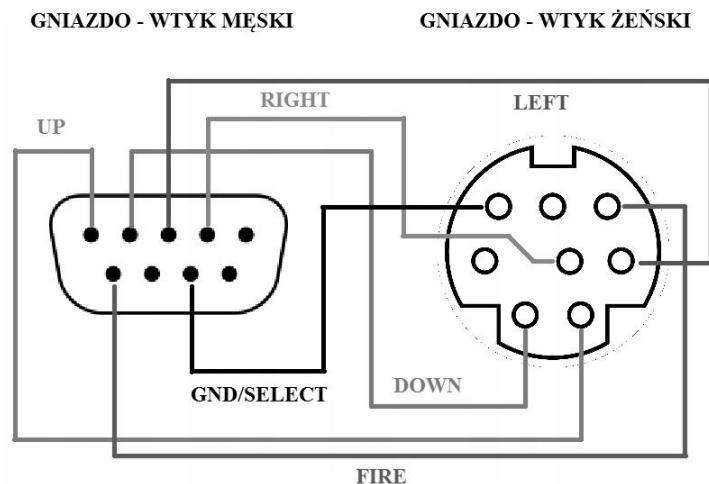
## Niesprawny egzemplarz i co dalej?

W moje ręce trafił niedawno Plus 4, który okazał się całkowicie niesprawny – objaw po podłączeniu to czarny ekran. Wpierw wymieniłem gniazdo zasilania, ponieważ w niektórych wersjach Plus'a 4 jest kwadratowe gniazdo – typowe wyłącznie dla tego modelu.

Jeśli posiadamy płytę o oznaczeniu 310163 to możemy bez problemu zamienić gniazdo na okrągłe DIN-7, dokładnie takie jak w C64. Pasują nawet dodatkowe otwory na płycie komputera. Wówczas do Plusa 4 możemy podłączyć standardowy zasilacz od Commodore 64.

Jedynie C16 i C116 używa innego wtyku – okrągłego jak do typowych zasilaczy impulsowych, gdzie doprowadzone jest napięcie 9VDC (uwaga na „+” zasilania, który jest na zewnątrz wtyku!).





Schemat podłączeniaa przejściówki joysticka.

Po wymianie kondensatorów i sprawdzeniu podstawowych rzeczy takich jak: bezpiecznik, napięcie, ewentualne zwarcia do masy (w Plus 4 o to bardzo łatwo), przerwane ścieżki (okazało się, że jedna taka się znalazła), a także po poprawie układów scalonych w podstawkach i sprawdzeniu ich temperatury, nic się nie zmieniło. Zacząłem sprawdzać więc wszystkie układy po kolei w innym sprawnym egzemplarzu. Zarówno procesor, jak i TED okazały się uszkodzone. Kernal, Basic i PLA były sprawne.

Pozostało przyjrzeć się pamięciom, ponieważ kilka z nich grzało się trochę bardziej niż pozostałe. Ostatecznie wymienilem wszystkie pamięci, a także kilka współpracujących z nimi układów scalonych. Po wstawieniu sprawnego TED'a i procesora komputer dalej nie wracał do życia. Kolejnym podejrzanym był układ resetu NE555 – ten okazał się być sprawny. Po wymianie kilku postów na forum miałem zamiar podmienić wszystkie niesprawdzone dotąd układy, które są podłączone do szyny danych i szyny adresowej.

Jednym z niewielu takich układów, które pozostały, okazał się być uszkodzony 20-nóżkowy scalak MOS 6529B (w Plus 4 są dwa takie) zarządzający także user-portem. Niestety jest on trudno dostępny, podobnie jak procesor czy TED, ale zamiast niego możemy użyć układu 74LS245, który pozwoli komputerowi działać, jednak user-port pozostanie niesprawny. Scalak 6529B na szczęście psuje się rzadko. Przy tak poważnych uszkodzeniach, które ciężko zlokalizować, w diagnostyce przyda się diag264:

<http://inchocks.co.uk/commodore/Diag264/HTMLManual/Diag264.htm>

Program testujący, który możemy wypalić w kości ROM, wstawić w miejsce kernala i próbować szukać uszkodzenia. Aby uruchomić komputer z tym programem diagnostycznym, musimy jednak posiadać przynajmniej sprawnego TED'a, procesor i PLA. Diag pozwoli na szybkie zdiagnozowanie problemu. Dość dokładna i bogata w informacje i porady o tym jak naprawić Plus'a 4 jest

instrukcja serwisowa:

<http://www.commodore16.com/plus-4c16-troubleshooting/>

gdzie krok po kroku jest opisane co należy sprawdzać w zależności od uszkodzenia, a przydatne na pewno okażą się też schematy wszystkich modeli:

<ftp://www.zimmers.net/pub/cbm/schematics/computers/plus4/index.html>

Na koniec, przy moim sprawnym już od strony elektroniki egzemplarzu, należało popracować nad klawiaturą, ponieważ działały jedynie 3 klawisze. Po rozebraniu okazało się że wystarczy oczyścić pola styku i same gumki przewodzące od spodu klawiszy – z pomocą przyszło IPA plus. Po oczyszczeniu klawiatura działa prawie jak nowa.

## Na zakończenie

Jeśli już posiadamy sprawnego Plusa 4 (wydaje się być on najrozsądniejszą platformą do poznania serii 264) lub opcjonalnie C16/C116 ale z rozszerzeniem pamięci 64KB – opis rozszerzenia znajdziemy tu:

[http://oldcomputer.info/hacks/c16\\_64k/index.htm](http://oldcomputer.info/hacks/c16_64k/index.htm)

bo tylko przy takiej konfiguracji jesteśmy w stanie poznać pełne możliwości tych sprzętów, to możemy zabrać się za jego eksploatację. Są bogate zasoby gier, wśród których możemy z pewnością znaleźć coś dla siebie:

<http://plus4world.powweb.com/home>

i dem, które znajdziemy tradycyjnie na pouet.net, czy demozoo.org. Na te platformy wciąż wychodzą nowe gry, zarówno w wersjach cyfrowych jak i na oryginalnych nośnikach:

<https://psytronik.itch.io>

czy chociażby:

<https://www.thefuturewas8bit.com>

Warto zainteresować się rodziną 264, poznać w działaniu i samemu przekonać się czy rzeczywiście komputery te były marketingową porażką, czy może rewolucją, która nie spotkała się z większym entuzjazmem użytkowników. Ja osobiście uważam, że te komputery, mimo że czegoś im jednak brakuje, mają ogromny potencjał. Gdyby tylko były mniej awaryjne i bardziej dostępne.

(\*) Dla ułatwienia, w tym artykule będziemy nazywać komputery Commodore C116, C16 i Plus/4 po prostu serią 264.

(\*\*) sprzętowy emulator stacji 1541 działający wyłącznie z Commodore C64, z którego jednak też możemy korzystać przy wczytywaniu danych do modeli C116/C16/Plus 4 – C64 w zestawie z osobnym monitorem staje się tutaj niejako „zasilaczem” i sterownikiem, emulowanej stacji. Po zamontowaniu obrazu dyskietki przewód serial podpinamy oczywiście do wejścia serial w serii 264.

**Podziękowania:** Rafał, Drygol, Bachoo, [forum64.de](http://forum64.de), AtariArea

**Przedstawione w artykule zdjęcia są własnością ich autorów.**

# Przyspieszamy PET-a

Opracował:  
**MARIUSZ  
WASILEWSKI**

Commodore PET wygląda tak klasycznie, że jego rozbudowa wydaje się wręcz barbarzyństwem. Jednak w tym artykule chcę powiedzieć jedynie o programowych metodach optymalizacji programów w Basicu.

Zwiększanie wydajności programów w Basicu nie musi wiązać się tylko z analizą działania algorytmów i tworzenia bardziej doskonałych programów. Czasem pomagają także drobne poprawki, które potrafią "zrobić różnicę". Można to wykorzystać na przykład w stosunku do podprogramów.

Jedną z takich prostych technik jest odpowiednie numerowanie linii, od których będą się później odwoływać polecenia GOTO i GOSUB. Głównym problemem przy ich wywoływaniu jest czas potrzebny na to, aby Basic odnalazł właściwy numer linii, który ma zostać wywołany.

Użytkownik tego nie obserwuje, ale porównywany jest bardziej znaczący bajt numeru linii docelowej z takim samym bajtem linii aktualnie wykonywanej. Jeśli docelowy bajt jest większy, Basic rozpoczyna wyszukiwanie od

następnej linii programu. W przeciwnym razie program przeszukiwany jest od początku listingu.

W związku z tym można wykonywać małe skoki "do przodu", co pozwoli przyspieszyć działanie podprogramów. Jak numerować linie, aby uzyskać taki skutek? Najlepiej obliczyć numer według poniższego wzoru:

```
Minimalny numer li-  
nii docelowej = 256 *  
{INT(numer bieżącej li-  
nii / 256) + 1}
```

Dzięki taki zabiegom można zwiększyć prędkość wywoływania podprogramów około trzykrotnie - wystarczy sama zmiana numeracji linii. Oczywiście nie oznacza to, że cały program będzie działał kilka razy szybciej. Ilość zaoszczędzonego czasu zależy bezpośrednio od długości programu i pozycji instrukcji GOTO lub GOSUB w programie.

# Simons' Basic

## Używanie sprajtów

KAMIL STOKOWSKI

Każdy użytkownik C64 wie, że jego maszyna potrafi generować sprajty. Jest to element ekranu, który może być kontrolowany niezależnie od pozostałej zawartości. Dane mogą być modyfikowane w celu przemieszczania sprajtów, zmiany koloru, a nawet zmiany kształtów bez konieczności zmiany "tła". W stosunkowo prosty sposób można zaprogramować nawet osiem równoległych i niezależnych sprajtów. Chcę pokrótce opowiedzieć, jak radzić sobie z obsługą takich elementów w popularnym i prostym w użyciu Simons' Basicu.

W Simons Basicu do obsługi sprajtów używamy poleceń mających w nazwie skrót MOB. Skąd to wynika? Otóż w naszym języku sprajt jest nazywany bardziej ogólnie - ruchomym obiektem (ang. movable object block), dlatego będziemy posługiwać się taką terminologią. Sprajty, czy MOB-y, mogą być wyświetlane zarówno na ekranie niskiej, jak i wysokiej rozdzielczości (Lores i Hires).

W trybie Hires można je wyświetlać w dowolnym kolorze, a w trybie wielobarwnym w maksymalnie trzech różnych kolorach. Sprajt w rozdzielczości Hires ma 24 piksele szerokości oraz 21 pikseli wysokości. Należy pamiętać, że w trybie wielokolorowym piksele mają podwójną szerokość, więc wielokolorowe MOB-y mają 12 punktów szerokości i 21 punktów wysokości.

Simons Basic posiada wiele specjalnych komend zaprojektowanych specjalnie po to, aby praca ze sprajtami była jak najprostsza. Pierwszym ważnym poleceniem jest DESIGN, którego zadaniem jest przydzielenie miejsca w pamięci dla MOB-a. Zanim wykonamy kolejne operacje, musimy określić rodzaj projektowanego sprajta oraz obszar, gdzie będzie on znajdował się w pamięci. Każdy sprajt wykorzystuje 64 bajty pamięci.

Instrukcja DESIGN posiada dwa parametry, wpisywane po przecinku. Pierwszy z nich określa, czy MOB ma być wyświetlany w trybie wysokiej rozdzielczości, czy w trybie wielobarwnym. Wartość 1 w tej pozycji oznacza tryb wielokolorowy, natomiast 0 - wysoką rozdzielczość. Drugi parametr określa adres początkowy pierwszego bajtu danych dla sprajta.

Liczba ta musi być wielokrotnością wartości 64 w zakresie od 2048 do 16319 i może być wprowadzona w postaci dziesiętnej lub szesnastkowej. Liczba szesnastkowa musi być, jak zwykle, poprzedzona znakiem dolara (\$). Jeśli MOB ma być używany na ekranie graficznym o wysokiej rozdzielczości, do tej liczby musimy dodać stałą wartość 49152, którą również można wpisać jako szesnastkową.

Każdy 64-bajtowy obszar dostępnej pamięci sprajta nazywa się blokiem. Jeśli podzielimy adres startowy MOB-a przez 64, otrzymamy numer bloku. Jest on używany przez kolejne polecenie MOB SET do ustawiania parametrów sprajta. Pamięć przypisana dla MOB-a zaczyna się od adresu 2048 i jest dalej używana w blokach po 64 bajty.

Polecenie MOB SET, jak zresztą sugeruje jego nazwa, inicjuje MOB-a. W tym wypadku musimy podać aż pięć parametrów, cały czas rozdzielonych po prostu przecinakami. Pierwszy parametr określa numer sprajta, którego chcemy używać. Numer ten musi być unikalny dla każdego MOB-a. Im niższy numer, tym większy będzie jego priorytet w stosunku do innych MOB-ów, tzn. jeśli dwa lub więcej sprajtów przemieszcza się po ekranie, MOB o niższym numerze przechodzi "nad" drugim, który posiada wyższy numer.

Drugi parametr polecenia MOB SET określa blok pamięci, z którego będą pobierane dane MOB-a. Następny parametr definiuje główny kolor sprajta, czyli kolor, który należy przypisać do każdego punktu narysowanego MOB-a. Następny parametr określa priorytet MOB-a w stosunku do innych danych na ekranie. Mówiąc prościej, ustawiamy tutaj czy MOB ma przesłaniać znaki na ekranie, czy też nie. Liczba 0 ustawia wyższy priorytet sprajta, natomiast 1 daje pierwszeństwo danych widocznych na ekranie przed MOB-em. Ostatni parametr wskazuje, czy MOB został utworzony w trybie wielobarwnym czy wysokiej rozdzielczości. Wartość 1 oznacza wielobarwny, a 0 - wysoką rozdzielczość.

Na przykład, poniższa linia:

**MOB SET 8,32,8,1,8**

spowoduje ustawienie wysokiej rozdzielczości MOB-a. Natomiast w celu skonfigurowania wielobarwnego MOB-a możemy użyć takiego wpisu:



#### **MOB SET 1,33,2,8,1**

Jeśli ograniczamy się do wyświetlania wielu sprajtów na raz, możemy chcieć zdefiniować o wiele więcej parametrów. Kiedy już to zrobimy, możemy je zmieniać "w locie". W ten sposób można animować MOB-y. Dzięki szybkiemu przełączaniu różnych sprajtów możemy sprawić, że ruch będzie wyglądał wiarygodnie, na przykład postać w grze może biegać lub skakać. To tylko jeden z powodów, dla których często należy przechowywać w pamięci znacznie więcej danych, niż osiem zdefiniowanych sprajtów.

Kolejne polecenie to @, które jest nagłówkiem wskazującym siatkę MOB-a, która ma być stosowana, czyli po prostu podział poszczególnych znaków na punktu widoczne na ekranie. Siatka ma szerokość 24 punktów w trybie Hires i 12 punktów podczas używania wielokolorowego

MOB-a. Podczas projektowania monochromatycznego sprajta w trybie Hires, znakiem kodowym używanym przy wypełnianiu punktów jest B. Będzie to kolor przypisany w poleceniu MOB SET. W ten sposób możemy zorientować się, jak będzie wyglądał nasz MOB. Dalej musimy użyć komendy CMOB, czyli "color MOB". Pozwala na zdefiniowanie dwóch dodatkowych kolorów, które będą używane dla wielokolorowego sprajta. Format polecenia jest bardzo prosty, podajemy tylko dwa parametry oznaczające dwa kolory MOB-a. Oba wybieramy z zakresu od 0 do 15. Oczywiście, te parametry to dwa dodatkowe kolory, których możesz potrzebować, aby sprajt wyglądał tak jak trzeba. Tak jak mówiliśmy, podstawowy kolor MOB-a jest określany w instrukcji MOB SET.

Przy projektowaniu wielokolorowych sprajtów na siatce używane są znaki

B, C i D. Jednak kody przypisania kolorów nie działają tak, jak można się ich spodziewać, ponieważ znak B reprezentuje teraz pierwszy parametr polecenia CMOB, znak C reprezentuje kolor przypisany w poleceniu MOB SET, a znak D reprezentuje drugi kolor w poleceniu CMOB.

Kolejne polecenie MMOB oznacza "moving MOB", czyli chodzi o nadanie ruchu. Pozwala na wyświetlenie sprajta w jednym punkcie ekranu, a następnie przeniesienie go w inne miejsce. Format instrukcji jest nieco trudniejszy do zrozumienia i wygląda tak:

**MMOB nr,x1,y1,x2,y2,a,b**

Pierwszy parametr określa numer MOB-a, który chcesz użyć do wyświetlania i poruszania. Parametry x1 i x2 są współrzędnymi punktu na

ekranie, w którym sprajt zostanie wyświetlony, zanim wykonamy ruch. Parametry x2 i y2 wskazują miejsce docelowe MOB-a. Jeśli nie chcemy zmieniać pozycji MOB-a, wystarczy użyć tych samych współrzędnych określających lokalizację miejsca początkowego i końcowego.

Parametr A odnosi się do wielkości MOB-a, którą można dzięki temu zmieniać. Wartość 0 powoduje, że sprajt jest wyświetlany w normalnym rozmiarze, liczba 1 rozszerza kształt poziomo do dwukrotnej szerokości, liczba 2 robi to samo, lecz pionowo, natomiast 3 powoduje rozszerzenie zarówno poziomo, jak i pionowo. Zawsze jest to dwukrotność wysokości lub szerokości.

Ostatni parametr B określa szybkość, z jaką MOB będzie się poruszać. Ta liczba musi być z zakresu od 1 do 255. Wartość 1 oz-

nacza największą prędkość, natomiast 255 - najmniejszą. Oto przykład:

**MMOB 8,8,8,8,185,70,3,28**

W przypadku pisania gier bardzo ważnym elementem jest też możliwość wykrywania kolizji sprajtów. Do tego celu służy polecenie DETECT, które "włącza" obsługę kolizji. Wystarczy wpisać liczbę 1 po nazwie instrukcji. Bardzo ważną sprawą jest fakt, że polecenie to musi być użyte dwukrotnie, ponieważ jest najpierw używane do oczyszczenia obszaru w pamięci komputera, która wskazuje, czy doszło do kolizji. Aby wyłączyć wykrywanie kolizji wystarczy wpisać samo słowo DETECT, bo program uzna, że wprowadziliśmy "zero".

Następne polecenie to MOB OFF, które powoduje wyłączenie wyświetlania sprajta - trzeba tylko wpisać numer MOB-a. Należy pamiętać, że nie oznacza to usunięcia sprajta z pamięci, można go później przywrócić w to samo miejsce lub zmienić jego pozycję albo kształt. Wszystko zależy od przeznaczenia naszego programu.

Obsługa sprajtów w Simons Basicu nie jest trudna, ale trzeba zrozumieć specyfikę programowania w tym języku. Dla mnie było to dość naturalne, a dodatkową zaletą jest fakt, że takie programy można pisać szybko i od razu widzieć efekt na ekranie. Fragmenty oryginalnej instrukcji do Basica można znaleźć nawet na Wikipedii, niestety sekcja ta nie jest zbyt rozbudowana. Dlatego postanowiłem podzielić się moją wiedzą z Czytelnikami magazynu RetroKomp..



# Testujemy dźwięk na Commodore 64

Opracował:  
**MARIUSZ  
WASILEWSKI**

Możliwości dźwiękowe C64 przewyższały układy wcześniej produkowanych komputerów. Tak rozbudowana kontrola była dostępna tylko na syntezatorach dźwięku. Przedstawiony poniżej program pozwala eksperymentować z dźwiękiem, a następnie słuchać rezultatów i modyfikować ustawienia. Można korzystać z maksymalnie ośmiu oktav oraz generatora szumu.

Podczas uruchamiania programu pojawi się lista aktualnych wartości, z widocznym kursorem. Jeśli chcemy zmienić wartość należy wpisać nową liczbę, a następnie nacisnąć klawisz RETURN. Jeśli nie chcemy zmieniać wartości, naciskamy tylko sam RETURN.

Klawisze T, S, i P odnoszą się do fali trójkątnej oraz piłokształtnej. Znak N oznacza generator szumu, a Q - zakończenie działania programu. Za każdym razem, gdy chcemy zobaczyć aktualne parametry trzeba użyć litery D. Po wprowadzeniu ostatniej wartości, dźwięk zostanie odtworzony z aktualnymi wartościami ADSR, po czym program zapętli się i powtórzy działanie.

Zapraszamy do własnych eksperymentów z dźwiękiem na najlepiej sprzedającym się komputerze domowym w historii!

```
5 DIM S0 (16, 8)
10 HF = 54273 : LF = 54272 : AD = 54277 : SR = 54278
   : W = 54276 : U = 54296 : HP = 54275 : LP = 54274
15 FOR I = 1 TO 8 : A(I) = 0 : NEXT : W$ = "S"
20 FOR O = 1 TO 8 : FOR N = 1 TO 16 : READ S0 (N, O) :
   NEXT : NEXT
25 FOR N = 1 TO 8 : READ D (N) : NEXT
100 PRINT "{CLEAR}" : CHR$(18) : "PULSE SETTING USED
   ONLY WITH PULSE WAVE " : CHR$(146)

102 PRINT
110 PRINT "WAVEFORM (T, S, P, N) = " ; W$
112 PRINT "VOLUME (1-15) = " ; A (1)
114 PRINT "OCTAVE (1-8) = " ; A(2)
116 PRINT "ATTACK SETTING (0-15) = " ; A(3)
118 PRINT "DECAY SETTING (0-15) = " ; A(4)
120 PRINT "SUSTAIN SETTING (0-15) = " ; A(5)
122 PRINT "RELEASE SETTING (0-15) = " ; A(6)
124 PRINT "HIGH PULSE SETTING (0-15) = " ; A(7)
126 PRINT "LOW PULSE SETTING (0-255) = " ; A(8)
130 PRINT "{HOME}" : "{03 DOWN}" :
140 PRINT TAB(33) : "Q" GOSUB 500 : IF Z$ = "D" THEN
600

150 IF Z$ = "Q" THEN PRINT "{CLEAR}" : END
155 IF Z$ < > CHR$ (13) THEN W$ = Z$
```



```

160 FORO = 1 TO 6
170 PRINTTAB(33) ; "?" ; : GOSUB500 : IF Z$ < > CHR$(13) THENA(0) = VAL(Z$)
180 NEXT
190 IF W$ < > "P" THEN 230
200 FORO = 7TO8
210 PRINTTAB(33) ; "?" ; :GOSUB500 : IF Z$ < > CHR$(13) THEN A(0) = VAL(Z$)
220 NEXT
230 POKEV, A(1)
240 POKEAD, 16 * A(3) + A(4) : POKESR, 16 * A(5) + A(6)

250 IF W$ = "T" THEN POKE W, 17
260 IF W$ = "S" THEN POKE W, 33
270 IF W$ = "P" THEN POKE W, 65 : POKEHP, A(7) : POKELP, A(8)
280 IF W$ = "N" THEN POKE W, 129

300 O = 0
310 FOR I = 1TO15 STEP2 : POKEHF, SO (I, A(2)) : POKELF, SO (I + 1, A(2)) : O = O + 1
: FORN = 1 TOD(O)
311 NEXT
315 POKEHF, 0 : POKELF, 0 : NEXT : POKEW, 0 : POKEAD, 0 : POKESR, 0 : POKEHP, 0
: POKELP, 0
320 GOTO 100

500 Z$ = " "
510 GETY$ : PRINTCHR$(18) ; " " ; CHR$(146) ; : FOR I =1TO25 : NEXT : PRINT
CHR$(157) ; " " ;
515 PRINTCHR$(157); : FORI = 1TO25 : NEXT : IFY$ = " " THEN510
520 PRINTY$;
530 IF Y$ = CHR$(13) THENIF LEN(Z$) = 0 THENZ$ = CHR$(13)
540 IF Y$ = CHR$(13) THEN RETURN
550 Z$ = Z$ + Y$ : GOT0510

600 PRINT "{CLEAR}"
610 PRINT : PRINT "ATTACK/DECAY = " ;16 * A(3) + A(4)
620 PRINT "SUSTAIN/RELEASE = " ;16 * A(5) + A(6)
630 PRINT : PRINT : PRINT "HIT ANY KEY TO CONTINUE" ;
640 GETY$ : IFY$ = " " THEN640
650 GOT0100

1000 DATA 1, 155, 1, 90, 1, 110, 1, 155, 1, 90, 1, 110, 1, 155, 1, 205
1010 DATA 3, 54, 2, 179, 2, 220, 3, 54, 2, 179, 2, 220, 3, 54, 3, 155
1020 DATA 6, 108, 5, 103, 5, 185, 6, 108, 5, 103, 5, 185, 6, 108, 7, 53
1030 DATA 12, 216, 10, 205, 11, 114, 12, 216, 10, 205, 11, 114, 12, 216, 14, 107
1040 DATA 25, 177, 21, 154, 22, 227, 25, 177, 21, 154, 22, 227, 25, 177, 28, 214
1050 DATA 51, 97, 43, 52, 45, 198, 51, 97, 43, 52, 45, 198, 51, 97, 57, 172
1060 DATA 102, 194, 86, 105, 91, 140, 102, 194, 86, 105, 91, 140, 102, 194, 115, 88
1070 DATA 205, 133, 172, 210, 183, 25, 205, 133, 172, 210, 183, 25, 205, 133, 230,
176
1080 DATA 500, 250, 250, 250, 250, 250, 250, 1000

```

# Rozbudowa pamięci w małym Atari

**MICHAŁ „stRing”  
RADECKI-MIKULICZ**

**Rozszerzenie, które tu opiszę właściwie w obecnych czasach może się wydawać nieopłacalne i niepraktyczne. Na rynku dostępne są przecież moduły Ultimate 1MB, które zapewnią naszemu Atari wszystko co potrzeba i kosztują stosunkowo niewiele (do pełnej, rozsądnej funkcjonalności brakowałoby jeszcze stereo. VBXE i Rapidus'a celowo pominę, bo ma swoich zwolenników jak i przeciwników, a i bez nich uruchomimy 99,9%**

Jednakże obecnie wielu kolekcjonerów ma nie jeden, dwa czy trzy modele, ale znacznie więcej i przychodzi taki moment kiedy chcemy sobie uruchomić produkcję, która wymaga więcej niż 64KB pamięci, ale musi się uruchomić na stockowym sprzęcie, ot taka dewiacja. Możliwość taką daje oczywiście posiadanie 130XE, które ma fabrycznie zamontowaną rozszerzoną pamięć, ale nie każdy może jeszcze posiadać taki model w swojej kolekcji, lub akurat on doczekał się przystawki w formie wspomnianego Ultimate 1MB i już nie jest taki fabryczny jak by się chciało.

Kolejnym powodem może być czysta przyjemność, bo przecież oprócz uruchamiania produkcji na swoich zabawkach, nie rzadko użytkownicy lubią w nich po prostu pogrzebać dla własnej satysfakcji. Ja należę właśnie do takich osób, stąd pomysł

na rozszerzenie pamięci, którego nigdy wcześniej nie wykonywałem.

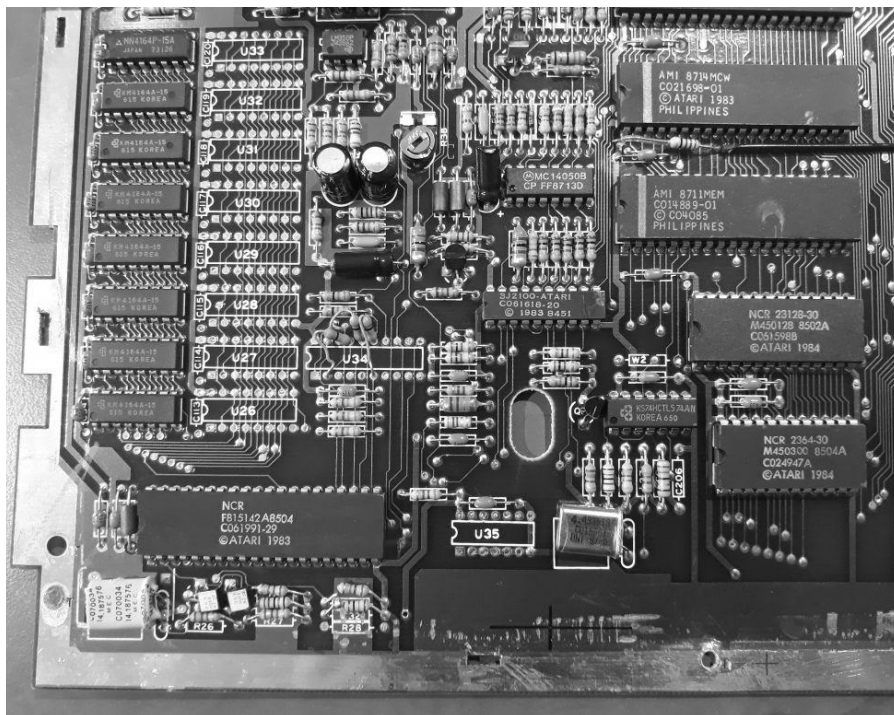
Wprowadzona za czasów Jacka Tramiela, w 1985 roku, nowoczesna w wyglądzie linia XE doczekała się trzech modeli mikrokomputerów: 65XE, 130XE i 800XE (wspomnijmy, że była też na rynku dostępna konsola XEGS). Wszystkie te trzy komputery, poza brakiem złącza ECI w niektórych egzemplarzach, są identyczne i zgodne ze sobą. Oczywiście 130XE ma oprócz standardowej pamięci, zamontowane fabrycznie dodatkowo 64KB.

We wszystkich modelach przewinęło się kilka wersji płyt głównych, a te najbardziej popularne to: C103579 (na nim budowane były modele 65XE, 130XE i 800XE) z miejscem na 2 (4) 4-bitowe kostki pamięci, C070067 (głównie 65XE i 130XE) z miejscem na 8 (16) 1-bitowych

kostek pamięci, oraz C070025 (model 65XE, wersja bez ECI), także na 1-bitowych kości pamięci, ale z miejscem na jedynie 8 kostek. W Polsce najbardziej popularny był 65XE, bowiem 130XE był droższy, a 800XE przeznaczony był głównie na rynek zachodni.

W moje ręce trafiła płyta C103579. Była ona zdekompletowana, brakowało elementów dyskretnych, niektórych układów scalonych, a także gniazd. Postanowiłem ją odratować, nabyłem więc potrzebne części i płytę przywróciłem do życia. Potem postanowiłem rozbudować ją do 128KB i na łamach tego artykułu postaram się opisać jak można tego dokonać.

Przed wszystkim płyta ta jest idealna do rozbudowy - ma miejsce na 2 dodatkowe 4-bitowe kości pamięci, co znacznie ułatwia pracę.



Oprócz pamięci, potrzebować jeszcze będziemy dodatkowego układu scalonego, mianowicie CO25953. Jest to układ zarządzający dostępem do pamięci, zwany małe MMU (od Memory Management Unit). Występuje on wyłącznie w komputerach 130XE i jest dość trudno dostępny. Można go na przykład odlutować z uszkodzonej płyty od 130XE.

Taki układ wstawiamy z kolei w miejsce oznaczone U34 na rozbudowywanej płycie, skrajnie z lewej strony (pin 1 musi się zgadzać z polem lutowniczym 1), ponieważ ma on jedynie 16 nóżek. Wpierw jednak musimy usunąć trzy zworki wyglądające jak oporniki, posiadające rezystancję bliską zeru. Tutaj także warto rozważyć użycie podstawki (w tym przypadku będzie to DIP-20 lub DIP-16 – jedynie przy oryginalnym MMMU). To załatwia sprawę całkowicie, w tym momencie mamy już 128kB na pokładzie i komputer dzięki wmontowanemu MMMU potrafi ją obsłużyć.

Co jednak jeśli nie posiadamy układu CO25953? Otóż jest i na to sposób. Możemy zakupić GAL16V8, programowalny 20-nóżkowy układ scalony i ze strony:

<https://www.atarimax.com/jindroush.atari.org/achemmu.html>

pobrać plik co25953-gal.jed, którym zaprogramujemy GAL'a, lub szukać już gotowych, zaprogramowanych układów na serwisach aukcyjnych, lub forach, gdzie czasami ktoś sprzedaje, bądź programuje takie układy na życzenie. Układ ten nie zastępuje w pełni układu MMMU i aby cieszyć się pełną funkcjonalnością

Są oczywiście sposoby na rozszerzenie pamięci w innych modelach, ale wymagałoby to więcej pracy, gymnastykowania, a także większej ilości elementów jakie musielibyśmy zużyć na rozbudowę komputera. Jeśli dysponowalibyśmy płytą C070067, to rozbudowa byłaby nawet łatwiejsza, chociaż samego lutowania i potrzebnych części byłoby więcej, o czym za chwilę.

Montując na płycie C103579 dodatkowe kości pamięci warto oczywiście zamontować w podstawkach 18 pinowych typu DIP (miejsca U11 i U12), aby można je było w przyszłości bez trudu podmienić. Niestety otwory montażowe pod dodatkowe kości są zwykle zalane cyną i usunięcie jej bez profesjonalnej maszyny (stacja lutownicza, lub rozlutownica) może przysporzyć nie lada kłopotu.

Pokuszę się nawet o stwierdzenie, iż jest to trudniejsze niż w przypadku wlutowanego układu scalonego.

Można oczywiście posłużyć się odsysaczem i zwykłą lutownicą kolbową, ale to wersja dla cierpliwych. Przed usunięciem starej cyny, warto nanieść na pola lutownicze odrobinę nowej cyny i roztopić mieszając ją ze starą.

Po oczyszczeniu pól i wlutowaniu nowych podstawek dolutowujemy jeszcze 2 kondensatory ceramiczne o pojemności 100nF (oznaczenie 104) obok kostek pamięci, w miejsca C36 i C37. Kości pamięci jakie zastosowałem mają oznaczenie OKI M41464-10, są to kości typu 64k x 4 bit, jedna ma 32KB pamięci. Oczywiście z powodzeniem można użyć innych zamienników jak np. SHARP LH2464, Texas Instruments TMS4464, czy podobne. Warto aby jedna z kości miała większy czas dostępu (120, lub 150ns), wówczas nie są podtrzymywane w pamięci dane zaraz po wyłączeniu i nie trzeba odczekać kilkunastu sekund przed ponownym włączeniem komputera.

cią należy zamontować jeszcze jeden układ scalony 74LS95. Osobiście nie sprawdziłem jak zachowa się komputer z zamontowanym wyłącznie GAL'em, jednakże z tego co udało mi się dowiedzieć to nie wszystkie programy (w tym i dema) będą działać poprawnie i tak przykładowo w demie Video Blitz z 1987 roku niepoprawnie wyświetli się jeden z efektów.

Niestety trzeba się trochę natrudzić przy montażu scalaka 74LS95, ponieważ na płycie C103579 nie ma dla niego dedykowanego miejsca, w przeciwieństwie do płyty C070067, o czym wspominaliśmy wcześniej. Układ ten należy zatem umieścić na dodatkowej podstawie i przewodami podłączyć brakujące sygnały.

Dokonując przeróbki na płycie C070067 i wykorzystując GAL'a a nie dedykowany układ CO25953 wystarczy wlutować go wraz z podstawką (miejsce U34), oraz 74LS95 (miejsce U35), a także 8 kości

pamięci w miejsca U26 do U33, wraz z kondensatorami o pojemności 100nF każdy, w miejsca C113-C120.

Sposób montażu 74LS95 na płycie C103579 przedstawia ilustracja. Wykorzystano tutaj uniwersalną płytkę jednostronną dociętą na odpowiedni wymiar i wykonano odpowiednie połączenia przewodami. Dołączyć należy także masę i zasilanie 5VDC, a także dolutować kondensator 100nF. Schemat połączenia przedstawiony jest na kolejnej ilustracji.

Korzystano tu ze schematu narysowanego przez pana Jerzego Sobolę, jednakże wprowadzono jedną poprawkę, mianowicie noga 9 nie ma być połączona z nogą 19 GAL'a (sygnałem Ø2) jak na oryginalnym schemacie, tylko z buforowanym sygnałem Ø2, który znajdziemy m.in. na 7 nodze układu POKEY (CO12294), lub bliżej, przy założeniu że montujemy dodatkowy scalak pod kośćmi pamięci, na 30

nodze układu GTIA (CO14889). Na płycie C070067 te połączenia już są obecne, tak samo jak kondensator ceramiczny i nie trzeba już nic dorabiać.

Istotną rzeczą jest także podłączenie 12 nogi GAL'a do masy (zarówno w przypadku płyty C103579, jak i C070067), inaczej rozszerzenie nie będzie działać. U mnie to połączenie zostało wykonane od spodu równolegle do przeciwległej nogi o numerze 9, jako że jest ona podłączona do masy fabrycznie. Można też wstawić przełącznik, bądź zworę pomiędzy te piny, aby zostawić sobie możliwość ustawienia trybu bez rozszerzonej pamięci.

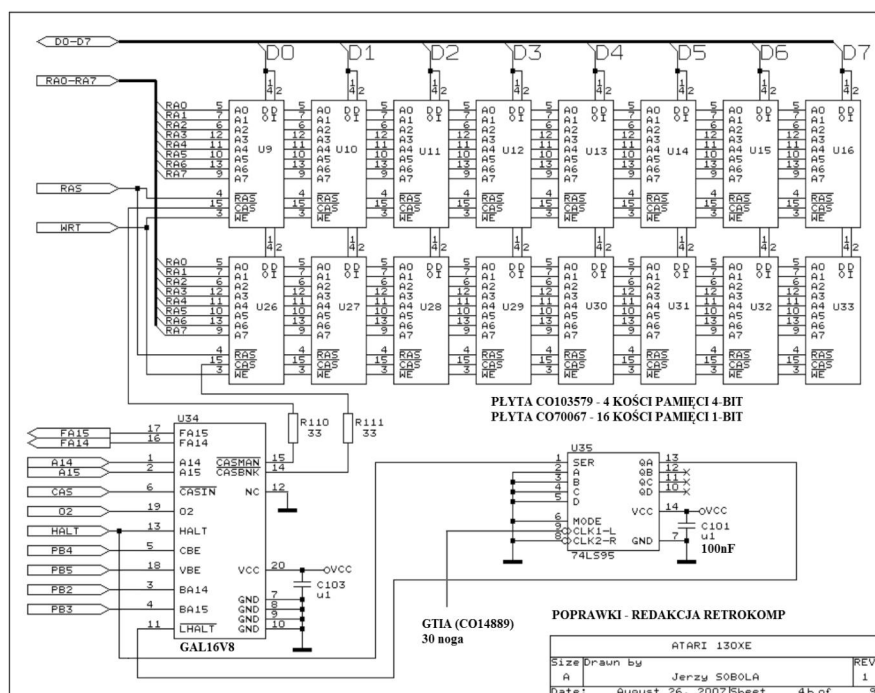
Na koniec warto się zastanowić co daje takie rozszerzenie? Jest trochę produkcji które korzystają z rozszerzonej o 64KB pamięci, zarówno starszych gier, jak i nowych. Wymienić można m.in. Bosconian, Laura (wersja kasetowa), Crowland, Klony 2010 (poprawiona wersja m.in. z obsługą myszki), Megablast, Karateka, Gato, The Goonies, Stunt Car Racer, The Eidolon i inne.

Pełną listę można znaleźć np. na stronie:

<http://www.mushca.com/f/atari/index.php?idx=0>

wpisując w wyszukiwarkę 130 XE. Kilka dem też uruchomi się z powodzeniem na 128KB, między innymi demo Near. Zatem lutownice w dłoń i powodzenia w rozbudowywaniu Waszych atarynek!

Podziękowania dla krystian77, serwis atari.org.pl.



# Wykopaliska Ninja



**PIOTR "KROLL"  
MIETNIEWSKI**

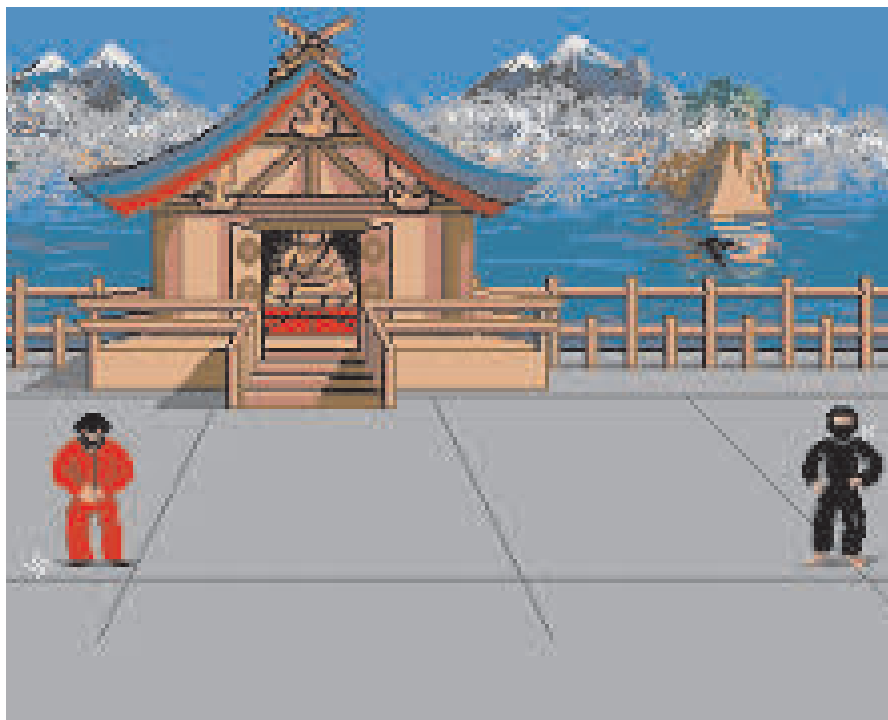
W numerze 11 RetroKompa rozpocząłem w tym dziale cykl artykułów opisujących gry na „duże” Atari, które powstały w oparciu o ich pierwowzory na Atari 8-bit. Pierwszą taką grą, którą odkopałem była omówiona wówczas wersja Miner 2049r na komputer Atari ST. Tym razem podczas odgrzebywania starych gier na moim STE trafiłem na klasykę, grę o znanym tytule na 8-bitowca a mianowicie NINJA.

Od niedawna ja, jako zagorzały fan dużych Atari zwłaszcza 32-bitowców, stałem się ponownie szczęśliwym posiadaczem komputera 16-bitowego. Mowa oczywiście o Atari STE. Jak większość osób, swoją ponowną przygodę z komputerami serii ST zacząłem od przypominania sobie starych dobrych czasów, czyli uruchamiania gier ze swojego dzieciństwa - jeszcze z czasów małego „Atari”. Do dnia dzisiejszego z wielkim sentymentem pamiętam jak wielokrotnie grałem i kończyłem grę o wspomnianym wyżej tytule. Na początek kilka słów o oryginalnym tytule na komputery 8-bitowe.

Autorem gry Ninja na Atari 8-bit jest Steve Coleman, znany również ze stworzenia takich tytułów jak Pharaoh's Curse czy Rainbow Walker. Grę Ninja wydał Mastertronic w 1986 roku. Autorem muzyki do gry Ninja jest Rob Hubbard - jednak nie przyznaje się on oficjalnie do jej atarowskiej wersji.

Gra Ninja powstała praktycznie na wszystkie popularne platformy 8-bitowe, czyli oprócz Atari na Commodore C-64, ZX Spectrum czy Amstrada. To tak dla przypomnienia,

każdy spokojnie może odszukać wiele artykułów w Internecie, wzmianek o tych wersjach i ich różnicach na poszczególne komputery 8-bitowe. Nie jest moim zadaniem



Przykładowy zrzut ekranu pochodzący z wersji gry na Atari ST.

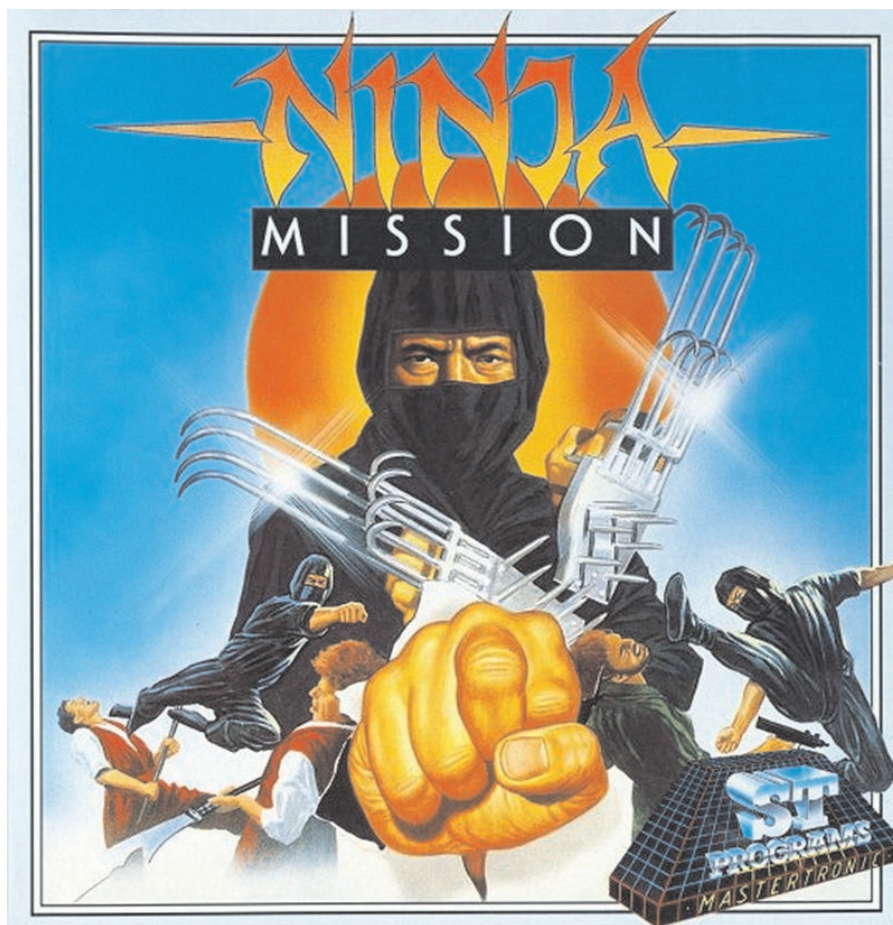
opisywanie różnic i wychwalanie wyższości jednej wersji nad drugą a jedynie przypomnienie jej pierwowзору z Atari XL/XE a przede wszystkim zaprezentowanie jej odpowiednika na Atari ST.

Samą historię możemy opisać w kilku zdaniach. Wydarzenia rozgrywają się w Perłowym Pałacu. Na jednym z pięter Perłowego Pałacu więziona jest piękna księżniczka, którą nasz tytułowy ninja musi uwolnić. W jej uwolnieniu przeszkadzają różni wojownicy posługujący się różnymi sposobami walki.

W drodze do odnalezienia więzionej księżniczki zbieramy posążki Idols, które uzupełniają nam energię, a także wskazują drogę do księżniczki. Wszystkich posążków jest 11. Do dyspozycji mamy miecz samurajski, noże do rzucania oraz shurikeny czyli tzw. gwiazdy ninja. Gdy trafimy przeciwnika ostrzem noża, to zabieramy mu więcej energii niż shurikenem. Przykładowe zrzuty ekranu z wersji na „małe” Atari przedstawiają rysunki 1 i 2.

Po krótkim wstępie i przypomnieniu sobie historii gry Ninja chciałbym przejść już do wersji tej gry na Atari ST. W artykule tym opisze tylko swoje własne wrażenia i odczucia, pragnąc zachęcić użytkowników ST, że warto czasami sięgnąć po stare tytuły. Wersja tej gry na komputer Atari ST ujrzała światło dzienne w roku 1987 pod lekko zmienionym tytułem jako „Ninja Mission”. Okładka tej wersji pokazana jest na rys. 3a i 3b.

Przy eksploracji swoich zasobów i odgrzebywaniu starych tytułów gier na Atari 16-bitowe moją szczególną uwagę przykuł właśnie katalog o



Okładka gry w wersji na Atari ST.

nazwie Ninja. Przypominając sobie jak wiele miłych wspomnień miałem z grą o takim tytule, z wielką ciekawością i zainteresowaniem uruchomiłem ją na moim STE. Jakże było moje ogromne zaskoczenie in plus, gdy przeszedłem już do samej rozgrywki, że bardzo ale to bardzo przypomina ona znany mi oryginał z małego Atari.

Pod względem grafiki - ze względu na większe możliwości graficzne komputerów serii ST - gra wygląda ładniej, jest więcej szczegółów na ekranie, do tego dochodzą fantastyczne tła upiększające tę grę. Przykładowe zrzuty ekranów z wersji na Atari ST są przedstawione na rysunku 4a i 4b.

To wszystko jest okraszone wspaniałą muzyką odzwierciedlającą klimat z platform 8-bitowych. Cała rozgrywka w tej grze w 100% oddaje klimat z pierwowзору. Mimo moich wcześniejszych uprzedzeń co do gier przenoszonych bezpośrednio lub pośrednio z wersji 8-bitowych na 16-bitowe, od samego początku stałem się ponownie jej wielkim fanem.

Reasumując należy powiedzieć, że „Ninja Mission” to konwersja klasycznej gry z 8-bitowych platform na komputer Atari ST i jako taka nigdy nie była przełomowa - i na pewno nie jest. Oczywiście, niewiele się zmieniło w stosunku do oryginału na komputer Atari XE/XL.



Osobiście cieszę się tą grą, szczególnie muzyką, którą możemy zaliczyć do najbardziej zapadających w pamięci melodii z jakichkolwiek gier z komputera Atari ST. Wysoka ocena tej gry, jaką przyznałem zawdzięczana jest przede wszystkim tym, że bardzo ale to bardzo dobrze oddaje klimat jej poprzednika z „małego” Atari, a lepsza grafika i muzyka tylko zwiększają jej atrakcyjność.

Oczywiście należy dodać, że „Ninja Mission” jest prostą grą bez jakiś wspaniałych, dodatkowych uroków,

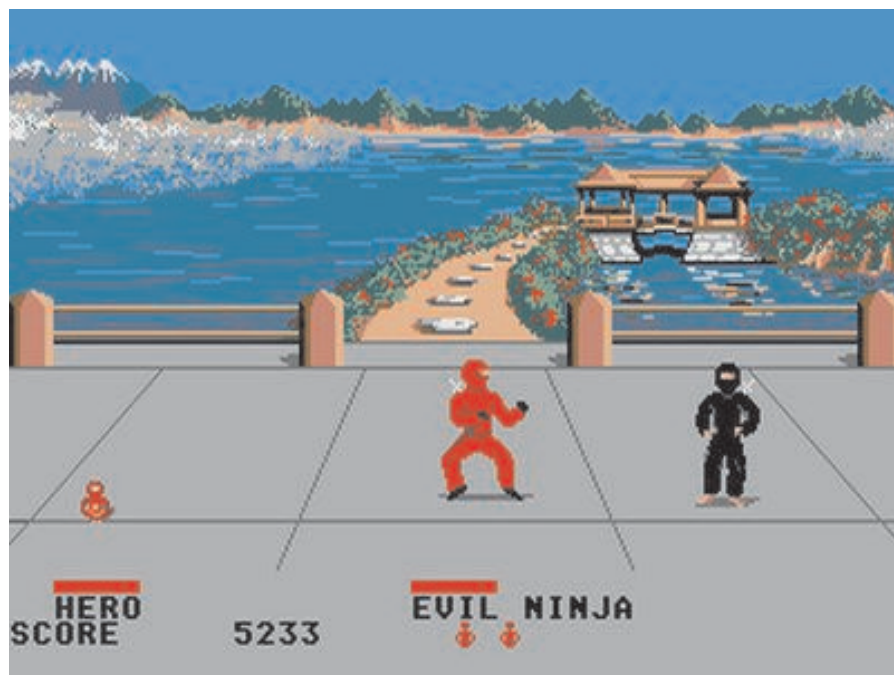


Zrzut ekranu z wersji gry na Atari XL/XE.

grafiki niewiele wnoszą, a bardzo często są pozbawione duszy, czyli tego co my cenimy sobie w grach od dziecka.

Podsumowując gorąco polecam tą grę każdemu użytkownikowi „dużego” Atari, a wspaniała muzyka i poprawiona grafika tylko mnie w mojej opinii utwierdziły. Wszystkich zainteresowanych, mających propozycje takich gier zapraszam bezpośrednio do kontaktu ze mną poprzez e-mail:

kroll@atarifan.pl



Przykładowy zrzut ekranu pochodzący z wersji gry na Atari ST.

która z całą pewnością nie jest kandydatem wagi ciężkiej na scenie gier Atari ST.

Artykuł ten stanowi kontynuację rozpoczętego w jednym z poprzednich numerów cyklu artykułów o grach bezpośrednio lub pośrednio przeniesionych na komputery Atari 16/32 bit, gdyż takich tytułów jest znacznie więcej, o których pewnie

niewielu słyszało, a warto odświeżyć i zobaczyć stare gry w nowej często lepszej oprawie muzycznej i graficznej. Przypominając w ten sposób swoim pociechom, czym to radowali się ich ojcowie. Ponadto porównując to, często do wszystkich wielu współczesnych tytułów, które mimo wspaniałej wręcz kosmicznej

#### Ocena:

grafika:



dźwięk:



grywalność:





# Starball

## Pinball w kosmosie

**PIOTR "Piter"  
KRUŻYCKI**

**Rok 1994. Polska zaczyna odbudowywać się po rządach komunistów. Kilka lat wcześniej zakończyła się tzw „zimna wojna” która mimo wszystko miała wg mnie klimatyczny wpływ na tematykę gier komputerowych w owym czasie.**

Rywalizacja dwóch wrogich obozów, NATO i Układu Warszawskiego bardzo często była wykorzystywana, w tym bardzo często w grach strategicznych. Jest to mój ulubiony rodzaj gier, a tytuły takie jak Theatre Europe, NATO Commander czy Conflict in Vietnam uruchamiały naszą wyobraźnię, wzbudzały zainteresowanie. Było to na Atari 8-bit i ładne kilka lat wcześniej.

Mamy 1994 rok, salony EMPIK, a w nich pojawiało się dość regularnie, tzn z regularnym opóźnieniem i dość drogie czasopismo tematyczne - prawdziwy powiew świeżości, w pełni kolorowy, wypełniony nowościami, opisami profesjonalnych programów, systemów operacyjnych i recenzjami gier. Mówię o kultowym ST Format! W numerze 64. tego angielskiego miesięcznika na cover dysku była umieszczona paczka z grą Starball, jej pełną wersją!

O tej grze słyszałem już wcześniej, ale do Polski wtedy jeszcze nie dotarła. Opisywana była w jednym z

poprzednich numerów ST Formatu i uzyskała ocenę aż 96%! Wróciłem z nowym numerem czym prędzej do domu i uruchomiłem na moim... no właśnie, nie pamiętam czy miałem jeszcze STe czy już miałem Falcona030. Wydaje mi się że byłem świeżo po zakupie Falcona, tak - jestem właściwie tego pewien, gdyż na tym komputerze gra miała dodatkowe możliwości, ale o tym napiszę trochę później.

Starball to pinball, typ gry która w tamtym okresie, w początkowych latach 90. miała swój renesans, głównie na Amidze ale nie tylko. Takie gry były również na STe i Falconie, a także komputerach PC (chodź sporo należało wydać, aby uzyskać podobną jakość jak A1200 czy Falconie 030). Gra przeze mnie opisywana powstała w 1994 roku i stworzona została przez grupę Volume 11.

Stół Starball podzielony jest na trzy sekcje. Pierwsza, gdzie na środku mamy wielkiego owada, robaka, po

prawej głowę obcego - sekcja ta nazwana jest organiczną. Tu toczy się walka o to, aby nie stracić piłki. Staramy się ukończyć ten etap, na końcu którego ukazuje się możliwość wbicia się w tułów owada!

Druga to sekcja kosmiczna, gdzie budujesz - tworzysz statek służący do podróży międzygwiazdnych, bronisz go przed atakiem wrogich statków i niszczysz działka próbujące uszkodzić Twój pojazd! Po ukończeniu również czeka Ciebie niespodzianka! Natomiast ostatnia, najwyżej osadzona sekcja to sekcja sił nadprzyrodzonych. Tutaj próbujesz nie dopuścić do przeobrażenia się potwora, niszcząc stworzenia krążące u góry ekranu.

W grze są także trzy bonusowe mini stoły, do których można dostać się po ukończeniu każdej sekcji. Jednym z takich stołów jest gra w Space Invaders, na innym musisz zniszczyć wielkie jajo, z którego odradzają się insekty. Kolorystyka wszystkich stołów jest wykonana z gustem, w

dobrym klimacie i cyber-kosmiczno-naukowym stylu. Piłka porusza się płynnie i jest zawsze widoczna, nie zlewa się z tłem.

Na szybszych maszynach (np. na Falconie) mamy animację - około 50 klatek na sekundę, na ekranie pojawić się mogą jednocześnie nawet 32 obiekty, tzw sprajty. Każdy z nich jest pięknie narysowany i animowany. Na maszynach powyżej 512 KB pamięci (gra sama rozpoznaje na jakim sprzęcie została uruchomiona i co może nam dzięki temu zaprezentować) mamy dodatkowe efekty np tzw SuperFlip, czyli ekran odwraca się i to w czasie rzeczywistym.

Flipery mają to do siebie, że wciągają gracza, który próbuje odnaleźć wszystkie ukryte przestrzenie na stole, uderzyć piłką we wszystkie ścianki, odkryć każdy sekret. Podobnie jest ze Starball, nie można się oderwać nie kompletując całości sekcji, zdobywając coraz to więcej

punktów i szukając kolejnych bonusów i ukrytych opcji.

Gra Starball, i tu Ciebie czytelniku zaskoczę, jest grą tzw Shareware. Nie jest to komercyjny produkt, ale z takim może swobodnie konkurować! Jak wspomina w jednym z wywiadów, grafik Andrew Pomianowski, który wtedy był studentem na uniwersytecie, autorzy dostali aż kilkadziesiąt rejestracji ich gry. Starball był typu shareware, więc oznaczało to także wpływ pewnej gotówki przy tej okazji.

Jeszcze podczas kodowania finalnej wersji, a po wypuszczeniu wersji demo myśleli, że zainteresowanie nie będzie takie wielkie i że będą musieli głodować w akademiku. Mimo to wpływy z rejestracji, która kosztowała z tego, co pamiętam 5 funtów, nie były na tyle duże, aby grupa kontynuowała produkcję kolejnych gier na ST/Falcon. Próbowali oni jeszcze przeportować ją na platformę PC, ale prace nad tą wersją

nie zostały ukończone. Na pochwałę zasługuje fakt, iż gra uruchamia się na każdym ST mającym 512 KB pamięci RAM, jednak im lepsza wersja naszego Atari, tym lepiej gra się prezentuje.

Najlepiej uruchomić ją na komputerze Atari Falcon030, gdzie możemy podłożyć swój własny soundtrack w jakości nawet 50 kHz, który będzie umilał rozgrywkę.

Na Sokole także mamy najlepszą animację piłki oraz przesuwanie się stołu, który nie jest mały, gdyż zajmuje powierzchnię około 204x800 pikseli! Bardzo dobrze gra się oczywiście także na Atari STe a w szczególności Mega STe!

Jest to prawdopodobnie, a wręcz jestem pewien, jest to najlepsza gra shareware na komputery 16/32-bit firmy Atari.

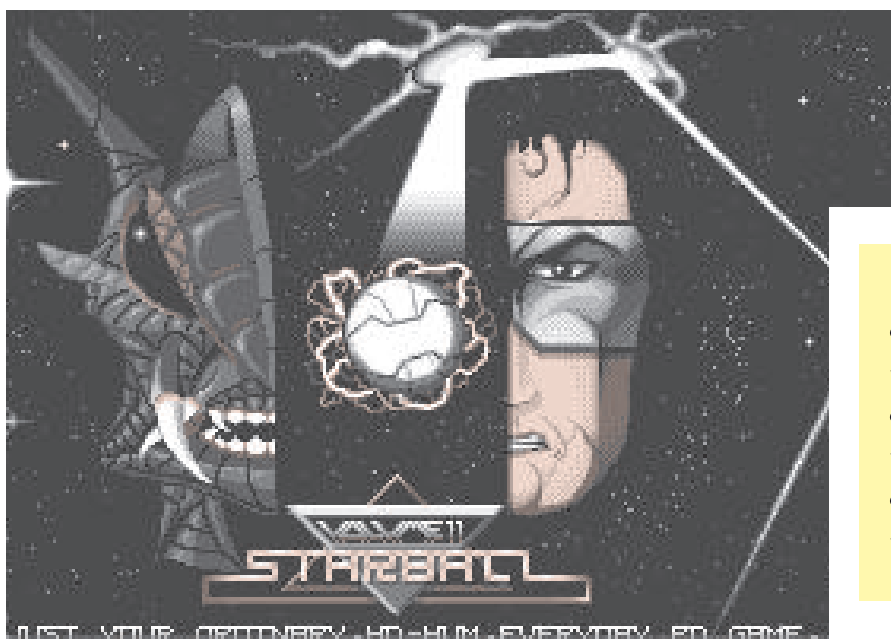
**Tytuł:** Starball

**Rok produkcji:** 1994

**Status gry:** Shareware

**Wykonana przez:** Volume 11

**Wymagania:** minimum ST 512 KB, specjalna wersja dla Atari STe/Falcon030



Bardzo ładny ekran początkowy gry.

#### Ocena:

grafika:



dźwięk:



grywalność:



# Rozszerzenia graficzne

MARCIN LIBICKI

**8-bitowe komputery Apple w latach swojej świetności były naprawdę rewolucyjne. Co prawda oferowały niewielką ilość pamięci RAM, ale posiadały bardzo dobrze opracowany język Basic, co wykorzystali hobbyści. Później czasy się zmieniły i inne komputery też mogły się pochwalić dobrą grafiką i dźwiękiem. Wśród nich były oczywiście Atari 800 i Commodore 64.**

Firma Apple traktowała jednak swoje produkty w trochę inny sposób. W modelu IIe skupiono się na funkcjach biznesowych, zamiast tworzyć jeszcze lepszą, kolejną maszynę do gier. Oczywiście kwintesencją komputera biurowego był model Lisa, ale wiele osób w okresie lat '80-tych mówiło o tym, że Apple powinno położyć większy nacisk na innowacyjność w zakresie możliwości audio-wizualnych.

Warto zauważyć, że firma nigdy nie starała się promować wszystkich możliwych zastosowań swoich produktów. Komputery Apple sprawdzały się bardzo dobrze w operacjach związanych z biznesem oraz zarządzaniem danymi. Z kolei producenci urządzeń peryferyjnych raczej kierowali się w stronę rozrywki, czyli na topie było polepszanie możliwości dźwiękowych czy generowanie sprajtów. Można powiedzieć, że firmy trzecie uzupełniały działanie Apple.

Tego typu rozszerzenia zmieniają sposób programowania i właśnie tutaj zaczyna się historia, którą chcę opowiedzieć. W pewnym momencie na rynku pojawiły się dodatki przekształcające komputery Apple w taki sposób, aby można było na nich mieć ulepszone funkcje związane z grafiką i muzyką, szczególnie jeśli chodzi o generowanie mowy i wspomniane już sprajty,

Dwoma firmami produkującymi takie rozszerzenia były Syntetic Systems i Third Millenium Engineering. W Polsce są to działania praktycznie nieznane, a te firmy naprawdę zmieniły obraz komputerów takich jak Apple II. Mam na myśli karty takie jak SuperSprite oraz Arcade Board.

Oba produkty zbudowane są na podobnej zasadzie. Procesor 6502 był bardzo dobry jak na czasy, w których powstał, ale w wielu wypadkach okazał się zbyt wolny do ob-

sługi lepszej jakości grafiki. Zamiast stosować sztuczki z optymalizacją programów czy pisaniem kodu maszynowego, firmy te zastosowały dodatkowy procesor graficzny przejmujący część zadań z jednostki 6502.

Jednym z częstą występujących problemów są zmiany kolorów w sąsiadujących pozycjach na ekranie. Gdy obiekt porusza się, powoduje to kolizję i zostaje wymazane tło. Nie powoduje to żadnych problemów, gdy jeden lub dwa obiekty poruszają się po czarnym tle. Jednak w sytuacji, gdy dziesięć obiektów przemieszcza się nad pomarańczowym tłem, można już widzieć, jak bardzo przeciętna jest grafika w Apple II.

Jako analogię tworzenie grafiki można przestawić jak w animowanej kreskówce. Postacie poruszają się po ekranie, przechodząc swobodnie na różnych planach. Mogą z łatwoś-

cią poruszać się w tle lub na pierwszym planie. Dzieje się tak dlatego, że obrazy są wielopłaszczyznowe - cały obraz składa się z wielu części, a każda część nie musi być przerysowywana przy każdym ruchu.

Grafika w Apple II jest wykonywana na jednym "planie", dlatego praca jest znacznie trudniejsza. Można maskować pewne obszary ekranu, ale jest to trudne, czasochłonne i w zasadzie daje niezadowalające rezultaty. Każdy obiekt musi być usunięty i narysowany na nowo przy każdym ruchu, co przy okazji zwalnia działanie programu.

Możliwość generowania sprajtów to większa szybkość, poza tym można je tworzyć bardzo łatwo. Nie trzeba stosować wspomnianych sztuczek. Zarówno SuperSprite jak i Arcade Board oferują 16 kolorów, które mogą być nakładane na siebie w celu uzyskania różnych odcieni. Dzięki 35 "planom" graficznym można narysować prawie każdy efekt.

W efekcie, zamiast przerysowywać cały ekran, wystarczy przesunąć tylko obiekt w jednej płaszczyźnie graficznej. Na przykład, jeśli mamy scenę samochodu jadącego przez pustynię, możemy animować zarówno płaszczyznę graficzną, na której narysowany jest samochód, jak i płaszczyznę tła. Są to jedyne elementy, którymi trzeba się przejmować. Pisząc program w "normalny" sposób, grafika musi być generowana na jednym "planie" i cała scena musi być ciągle rysowana na nowo, aby zachować płynność na ekranie.

Te możliwości można programować przy użyciu Basica i dołączonego

rozszerzenia, które wywołuje nowe funkcje graficzne. Jest to tylko dodatek, więc mamy zachowaną kompatybilność z Applesoft. Dzięki tym produktom, programowanie grafiki i sprajtów jest prawie tak samo proste, jak używanie zwykłych możliwości graficznych Apple II.

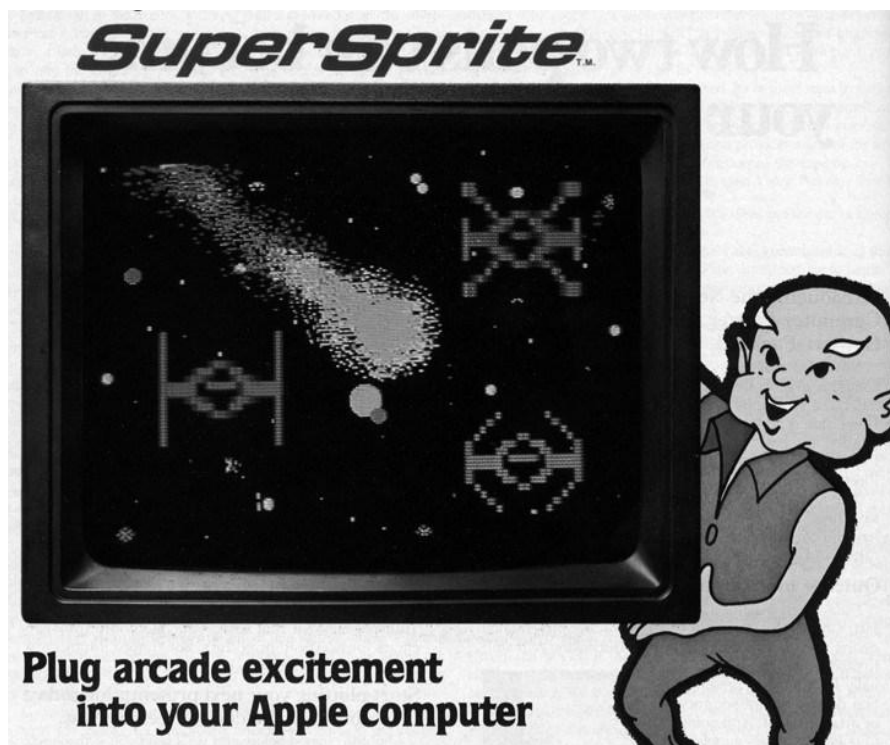
Karta SuperSprite składa się z procesora wyświetlania wideo Texas Instrument TMS 9918A i generatora dźwięku General Instruments AY3-8912PSG. Dodatkowo posiada 16 kilobajtów pamięci RAM do użytku przez funkcje graficzne oraz syntezator mowy. Pakiet zawiera oprogramowanie o nazwie Star Sprite, oprogramowanie Echo II Speech, głośnik, kable i dokumentację.

Oczywiście wiąże się to też z instalacją w komputerze. Na karcie należy podłączyć cztery wejścia. Jedno z nich jest podłączane do monitora, który umożliwia wyświetlanie normalnego obrazu, gdy karta Super-

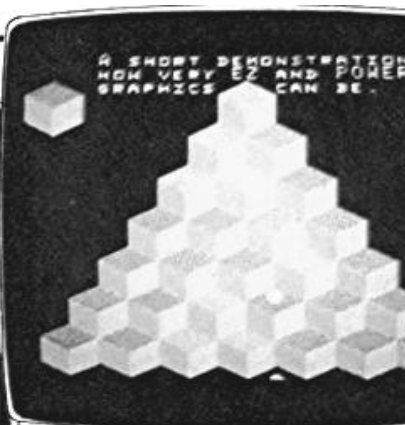
Sprite nie jest używana. Kolejny kabel jest łączony pomiędzy kartą i wyjście obrazu w komputerze. Trzecie połączenie jest między kartą i monitorem, który będzie używany podczas generowania sprajtów, a czwarte wejście/wyjście łączy kartę SuperSprite z głośnikiem.

Istnieją trzy różne pakiety oprogramowania dla karty SuperSprite. Star Sprite to pakiet dla początkujących. Star Sprite II i III to bardziej zaawansowane wersje, oferujące więcej narzędzi i informacji na temat programowania w języku maszynowym.

Wszystkie dyskietki są niezabezpieczone przed kopiowaniem, co pozwala na dostosowywanie programów do własnych potrzeb. Oprogramowanie jest bezużyteczne bez karty SuperSprite, więc producent nie musiał tworzyć zabezpieczeń, tak jak wielu innych wydawców w tamtym okresie.



## SuperSprite/Arcade, continued...



*Two sprite scenes. Arcade Board and center; SuperSprite, right.*

is because these images are *multi-plane* — the entire image is composed of many parts, and each part does not have to be redrawn with each move.

Because Apple graphics are done on a single plane, they are much more difficult

### SuperSprite

SuperSprite consists of the TMS 9918A video processor, the General Instrument 8912PSG sound generator, 16K of video use, and the Echo II

układy na rozszerzeniach są przeznaczone do wykonywania tylko określonych zadań, wykonują je znacznie lepiej i szybciej. Na przykład, aby przenieść postać o szerokości 40 pikseli, normalnie komputer Apple II musi narysować postać, wymazać ją i narysować na nowo w nowym miejscu. Aby to zrobić, potrzeba dosłownie tysięcy instrukcji w języku maszynowym. Przy użyciu SuperSprite lub Arcade Board, ta sama operacja może być wykonana podczas wykonania około dziesięciu instrukcji.

To samo dotyczy generowania dźwięku. W czasie odtwarzania nie można wykonywać innych operacji, dlatego gry mają zazwyczaj dobrą grafikę i tylko kilka efektów dźwiękowych.

Instrukcje używane na obu kartach są dość proste w użyciu w zwykłym programie w Basicu. Wystarczy używać zwykłych poleceń graficznych Applesoft, a w celu korzystania ze sprajtów i efektów dźwiękowych, należy wpisać instrukcje należące do rozszerzenia Basica.

Wykonywanie tych samych operacji przy pomocy nierozszerzonego oprogramowania Applesoft wymagałoby dużo bardziej rozbudowanego programu. W pierwszym przykładzie trzeba napisać program do generowania postaci w wysokiej rozdzielczości, czyli użyć tablic. W drugim przykładzie trzeba wykonać serię instrukcji POKE z odpowiednimi wartościami.

Możliwości programowania są ekscytujące. Wyzywające gry i innowacyjne programy są teraz w zasięgu ręki każdego właściciela Apple.

Oprogramowanie składa się z języka AmperSprite, programu do pomocy przy instalacji i konfiguracji, trzech krótkich gier i samouczka. W zestawie znajdują się również programy do tworzenia sprajtów, ich rysowania, tworzenia scen, generowania dźwięku i generowania tekstu. Wymagane jest niewielkie doświadczenie w programowaniu. W praktyce, jeśli ktoś programował w Basicu, poradzi sobie bez problemu z AmperSprite.

Jak już wspomniałem, kartę SuperSprite można było kupić z opcją syntezatora mowy, co pozwala dodawać ludzką mowę do swoich programów. Oprogramowanie Echo II umożliwia tworzenie słów przy użyciu fonemów, czyli najmniejszych rozpoznawalnych jednostek dźwiękowych języka.

Korzystanie fonemów pozwala na większą kontrolę nad mową, dzięki czemu brzmi ona bardziej naturalnie.

Warto dodać, że syntezytor mowy Echo Speech może być używany z grafiką sprajtów w ramach zwykłych programów Applesoft.

Karta Arcade Board wykorzystuje ten sam procesor wideo, co SuperSprite. Układem używanym do generowania dźwięku jest natomiast General Instruments AY-3-8910, czyli jednostka zbliżona do tej, który znajduje się w SuperSprite.

Pakiet zawiera oprogramowanie, ale nie ma podanej instrukcji na syntezę mowy, ponadto nie były dołączane przewody połączeniowe. Oprogramowanie składa się z rozszerzenia języka Applesoft i programu demonstracyjnego. Jego zastosowanie jest podobne do rozszerzenia Star Sprite na karcie SuperSprite.

Obie karty zdejmują z procesora 6502 całą pracę związaną z tworzeniem grafiki i dźwięku. Ponieważ

Ogromna znajomość języka montażu nie jest potrzebna; programowanie może być wykonywane przez Apple-soft z rozszerzeniami języka sprite.

Czy można powiedzieć, która karta jest lepsza? Pakiet SuperSprite oferuje syntezator mowy Echo i lepsze oprogramowanie. Oprogramowanie Star Sprite jest przeznaczone dla szerokiego grona użytkowników, od początkujących do profesjonalnych programistów. Karta Arcade Board ma mniej dodatków, ale kiedyś był też znacznie tańsza. Karta SuperSprite była też produkowana w kilku różnych wersjach, ale z tego co wiem różniły się tylko brakiem układu dźwiękowego.

Oprogramowanie obu produktów jest niekompatybilne. Program napisany dla SuperSprite nie może działać na płycie Arcade i odwrotnie. Nie można się temu dziwić, były to przecież konkurencyjne karty, ale dzisiaj może być tu denerwujące.



Jeśli będziecie mieli możliwość zapoznać się z obsługą sprajtów na Apple II, warto porównać je z tym, co oferuje na przykład Commodore 64. Opisywane karty nigdy nie uzyskały większej popularności, ale dzisiaj byłoby świetnie zobaczyć ich wykorzystanie przez scenowców.

# Zmiana nazw poleceń DOS

MARCIN LIBICKI

**Zapamiętywanie nazw i formatów poleceń DOS może być męczące. W wielu systemach operacyjnych można tworzyć tak zwane aliasy, czyli alternatywne nazwy instrukcji. Co można zrobić na 8-bitowym Apple?**

Program, który przedstawiam zmienia polecenia DOS. Po uruchomieniu wyświetla kolejne polecenia, po czym wyświetla komunikat. Jeśli chcemy zmienić komendę, wpisujemy tylko swoją wersję. Aby pozostawić komendę niezmodyfikowaną, naciskamy po prostu klawisz RETURN. Zostanie wtedy wyświetlone następne polecenie DOS.

Należy zachować ostrożność przy zmianie poleceń. Przede wszystkim, nie powinno się wpisywać nowych poleceń tak, aby pasowały do początków nazw innych poleceń. Na przykład, jeśli zmienimy "CATALOG" na "C", pojawią się problemy w sytuacji, gdy spróbujemy użyć poleceń CLR, CLOSE lub innych komend rozpoczynających się od litery C.

Po drugie, w poleceniu nie można umieszczać znaków Spacji, bo powoduje to nieprzewidywalne

zachowanie komputera. Komendy nie powinny być też zbyt długie ze względu na ograniczenia pamięci. Mogą mieć jednak inną długość niż oryginalne odpowiedniki.

Program będzie działał na DOS 3.3 z 48 KB pamięci RAM. Nie będzie działać z mniejszą ilością pamięci bez zmiany adresów w liniach z instrukcjami POKE. Po uruchomieniu zmiany mogą być zapisywane jako stała część DOS-u. W tym celu trzeba włożyć czystą dyskietkę i wpisać polecenie INIT oraz odpowiednie parametry.

Gdy inicjalizacja zostanie zakończona, dyskietka będzie miała zapisane modyfikacje systemu DOS. Później trzeba tylko uruchomić DOS z tak przygotowanego nośnika.

Dzięki tym zabiegom obsługa poleceń DOS może być łatwiejsza, a użytkownik może stworzyć własne środowisko pracy.



```

0 ON ERR GOTO 1000
1 DIM C$(28)
2 P = 1
10 FOR X = 43140 TO 43271
20 C$(P) = C$(P) + CHR$ (
PEEK (X))
31 IF PEEK (X) > 128 THEN
P = P + 1
32 IF P = 28 THEN 70
40 NEXT
70 HOME
80 PRINT "DOS CHANGER BY
ROBERT A. SWIRSKY"
160 VTAB 24 : PRINT "HIT
"C" TO CONTINUE" ; : GET
R$: IF R$ < > "C" THEN 160
170 HOME
180 PRINT : PRINT "YOU WILL
SEE ALL 28 DOS COMMANDS"
190 PRINT "IF YOU WISH TO
LEAVE IT UNCHANGED, PRESS
RETURN"
200 PRINT "TO CHANGE THE
COMMAND, SIMPLY RETYPE IT
WITH YOUR CHOICE."
210 PRINT : PRINT
220 LL = 43139
221 HL = 43271
230 FOR Q = 1 TO 28
235 PRINT C$(Q);
240 PRINT TAB( 20) ; """" > "
;
250 INPUT R$
260 IF R$ = "" THEN R$ =
C$(Q)
261 IF LEN (R$) = 1 THEN
PRINT "PLEASE "" TWO OR
MORE LETTERS" : GOTO 240
270 FOR W = 1 TO LEN (R$)
"1
280 POKE LL + W, ASC ( MID$
(R$, W, 1))
290 NEXT
295 IF ASC(RIGHT$(R$, 1)) >
127 THEN POKE LL + W, ASC
(RIGHT$(R$, 1)) : GOTO 501
300 POKE LL + W,
ASC(RIGHT$(R$, 1)) + 128
501 REM
510 LL = LL + W
520 IF LL > HL THEN 600
530 W = 0
599 NEXT
600 PRINT "CHARACTER LIMIT
EXCEEDED"
1000 PRINT "RUN TERMINATED"

```

# Cyfrowy termometr

Opracował:  
**MARIUSZ  
WASILEWSKI**

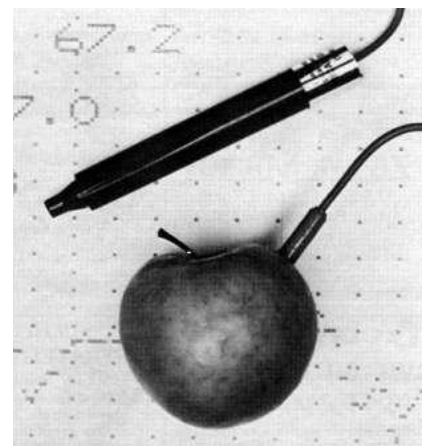
Na komputerze Apple II można mierzyć temperaturę! Mówiąc bardziej dokładnie, w latach '80-tych firma Americal Data Cable wyprodukowała zestaw dumnie określany jako "system pomiaru i kontroli temperatury".

Urządzenie pozwalało mierzyć temperaturę w zakresie od -60 do +105 stopni Celsjusza, a także uzyskać dokładności 0,01 stopnia. Całość nazwana "Nth Degree" wyglądała jak długopis z przewodem, a oprogramowanie działało w systemie DOS 3.3. Odczyty były wyświetlane w stopniach Celsjusza, Farenheita lub Kelwina. Dołączony program opcjonalnie mógł także pokazywać na ekranie lub drukować na bieżąco ciągły zapis zmian temperatury.

Czujnik mierzył szybkość reakcji zmiany temperatury, w związku z czym producent polecał go do monitorowania aparatury grzewczej i chłodniczej - jakkolwiek dzisiaj dziwne może się to wydawać. Program miał opcję pobierania danych raz na godzinę lub raz na sekundę przez 36 godzin. Ponadto można było programować powiadomienia w formie alarmów pojawiających

się na ekranie przy ustalonych wartościach. Gdy czujnik nie miał kontaktu z materiałem, którego temperaturę miał mierzyć, mógł sprawdzać temperaturę powietrza.

Urządzenie można uznać za ciekawostkę, ale ten przykład pokazuje, że komputery 8-bitowe były traktowane również jako narzędzie pracy. Dzisiaj możemy je też wykorzystać do prostych czynności, choć mało kto o tym myśli. Może wspomniany czujnik temperatury będzie dla Was inspiracją dla własnych pomysłów?



# Programowanie w edytorze tekstu

KAMIL STOKOWSKI

Pisanie kodu programu w Basicu na komputerze Apple odbywa się zazwyczaj przy użyciu oprogramowania Applesoft lub Integer Basic, które zapewniają podstawowe funkcje edycji oparte na swobodnym ruchu kursora oraz funkcji nadpisywania. Tych kilka narzędzi jest z pewnością odpowiednie do pisania programów, ale nie zawsze są najwygodniejsze. Jeśli nasz zestaw zawiera dysk twardy lub opieramy się na emulacji, możemy uruchomić edytor tekstu, aby wszystko stało się łatwiejsze.

Bezpośrednią zaletą jest ogromny wzrost komfortu programowania i czytelności listingu. Możliwe jest również, że w edytorze będziemy rozwijać w nieco inny sposób programy napisane wcześniej "normalnie". Dobry edytor tekstu może zapisywać kod programu Basica z kilku plików do nowego listingu i w ten sposób zapewnić alternatywę dla łączenia plików w Basicu. Jednym z takich starych, ale dobrych edytorów jest Magic Window. Można go bardzo dobrze wykorzystać także podczas programowania w Basicu.

Aby było to możliwe, zasadniczym wymogiem jest to, że edytor tekstu musi w stanie załadować, edytować i zapisywać standardowe sekwencyjne pliki tekstowe, które nie mogą zawierać żadnych innych znaków kontrolnych niż podstawowe znaki wstawiane w programie. Jest to możliwe w przypadku korzystania z

Magic Window poprzez wykonanie opcji FILER i UNFORMATTED LOAD/SAVE. Niektóre edytory tekstu nie przetwarzają plików jakie potrzebujemy, a niektóre wstawiają polecenia formatowania. Użytkownik, który posiada już gotowe listingi programów Basica, nie musi się martwić, gdy nowy kod zostanie zapisany w plikach tekstowych, ponieważ można je bardzo łatwo przenosić pomiędzy edytorem, a interpreterem Basica.

Jedną z nieprzyjemnych rzeczy związanych z używaniem Apple do pisania programów w Basicu jest fakt, że po wpisaniu linii programu i naciśnięciu klawisza RETURN, interpreter usuwa wszystkie "niepotrzebne" puste miejsca, a następnie zapisuje linię w pamięci w postaci zmodyfikowanej. Kiedy linia jest wyświetlana później, Basic najpierw wstawia puste miejsca zgod-

nie z własnym wzorcem, a następnie wyświetla ponownie "pełną" wersję linii programu. W związku z tym, tak utworzone linie programu rzadko przypominają te, które wpisywaliśmy wcześniej. W programie Magic Window tak się nie stanie, ponieważ to, co wpisujemy zostanie zapisane dokładnie, bez żadnych zmian.

Kolejną pozytywną cechą pisania programów Basic w edytorze tekstu jest to, że 80-znakowe linie mogą być zapisywane i drukowane na drukarce. Procesor tekstu będzie również zachowywał marginesy i pozwoli na wykorzystanie pustych linii do tworzenia struktury akapitów bez potrzeby używania instrukcji REM. Posiadanie 40 dodatkowych kolumn do pracy daje dodatkową pomoc i może być prawdziwą zachętą do pisania kodu i tworzenia przejrzystych wyglądających listingów.

Największym problemem związanym z programowaniem w Apple Basic może być to, że ani Applesoft ani Integer Basic nie zawierają bezpośredniego sposobu łączenia listingów zapisanych w dwóch lub większej ilości plików. Załóżmy na przykład, że programista ma trzy przetestowane i podprogramy zapisane w trzech różnych plikach, a nowy algorytm może być utworzony poprzez wykonanie tych podprogramów po kolei. Wczytywanie plików w czasie uruchamiania jest niezbyt dobrym podejściem do problemu, ponieważ może być wymagana tylko część każdego pliku. Poza tym jeżeli kontrola będzie przekazywana pomiędzy trzema sekcjami kodu wiele razy, może być to konieczne do wykonania w nieprzewidywalnej kolejności.

Najszybszym rozwiązaniem tego problemu może być zachowanie wszystkich gotowych podprogramów w plikach tekstowych i użycie edytora tekstu do przeniesienia wszystkich elementów potrzebnych do wykonania listingu do jednego nowego pliku tekstowego. Można to łatwo zrobić za pomocą polecenia INSERT FILE dostępnego właśnie w Magic Window.

Gdy wszystkie potrzebne fragmenty kodu zostaną przeniesione do nowego pliku, mogą być też łatwo ponownie ponumerowane lub porządkowane w zależności od potrzeb za pomocą poleceń CHARACTER-EDITING i LINE-EDITING. Służą one do wstawiania lub usuwania znaków, a także wstawiania, usuwania lub dublowania linii. Dalej elementy podprogramu mogą być połączone ze sobą za pomocą odpowiedniego kodu za pomocą sz-

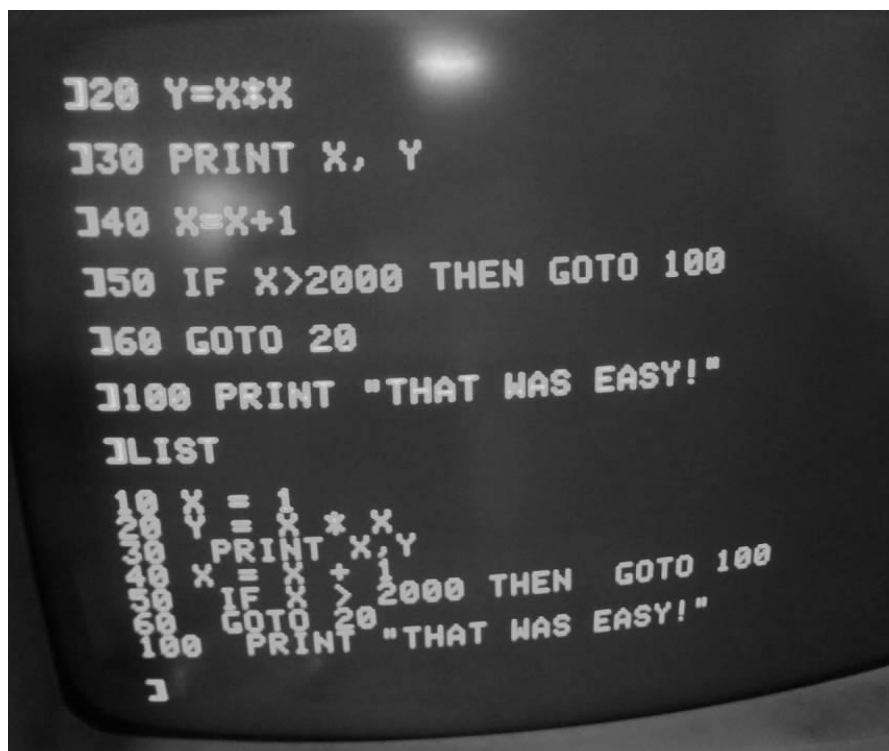
eregu instrukcji GOSUB, aby wywołać je w określonej kolejności.

Podczas używania edytora tekstu do edycji programów w Basicu, wszystkie litery, które nie są częścią instrukcji REM muszą być podane wielkimi literami. W programie Magic Window można zablokować klawiaturę w takiej pozycji, aby wszystkie kolejne znaki były wielkie. Można to zrobić przez dwukrotne naciśnięcie klawisza ESC w trybie EDITOR, albo przez wejście do CONFIGURATION i ustawienie opcji UPPER CASE ONLY na YES. Jeśli w instrukcjach REM potrzebne są małe litery, należy użyć sposoby z naciśnięciem klawisza ESC. Wielkie litery mogą być zablokowane lub odblokowane w dowolnym momencie w sekcji EDITOR.

Kolejną rzeczą, o której należy pamiętać jest to, że każda linia kodu w Basicu musi zaczynać się od numeru linii. W Applesoft linia może

mieć do 239 znaków i jest zakończona znakiem klawisza RETURN. Kiedy używasz interpretera, jedna długa linia może znaleźć się w kilku 40-znakowych liniach na ekranie. Jednak w Magic Window, długość linii jest ograniczona do 80 znaków i nie jest możliwe jej przeniesienie do kolejnego wiersza. Dlatego należy pamiętać, że każda linia musi być umieszczona w jednym wierszu o długości do 80 znaków. Edytor zachowa wszystkie puste linie, tak więc nasz listing może mieć elegancki wygląd.

Magic Window zapisuje listingi na dysku w plikach tekstowych, natomiast Applesoft nie może uzyskać do nich dostępu za pomocą polecenia LOAD. Mamy jednak polecenie EXEC powodujące, że numerowane linie programu są ładowane do pamięci, zaraz po liniach już wprowadzonych. Dlatego dobrym pomysłem jest wpisanie instrukcji NEW przed użyciem EXEC, jeśli nie chcesz, aby linie

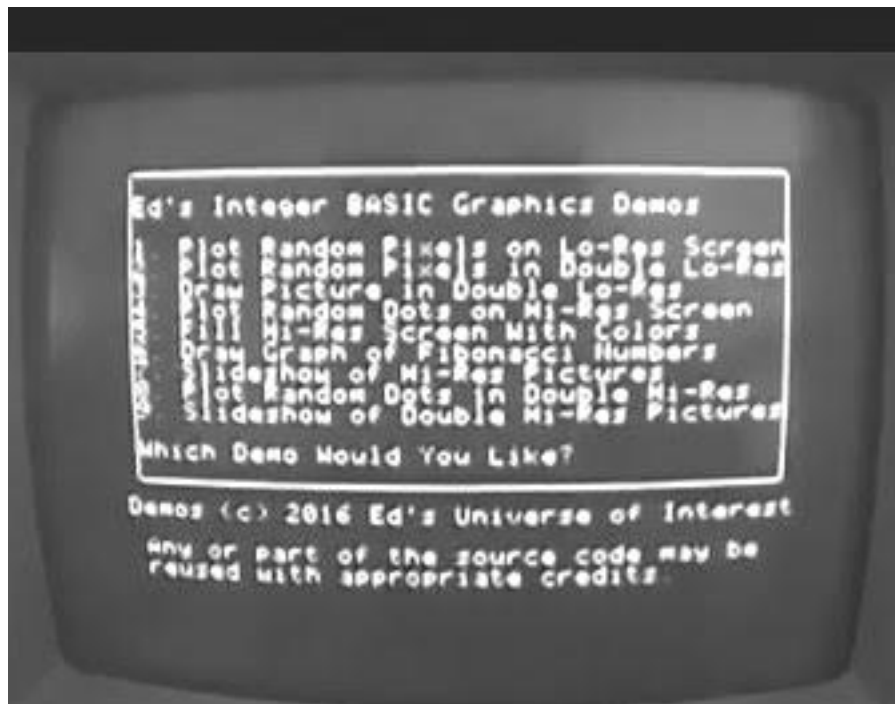


aktualnego programu nie mieszały się z liniami pochodzącymi z pliku tekstowego.

Możliwość wprowadzenia poleceń trybu bezpośredniego w pliku zawierającym program jest kolejną zaletą używania plików tekstowych. Może być to wygodne, ale nie jest w żaden sposób konieczne. Jeśli plik zawiera tylko numerowane linie programu, polecenie EXEC ładuje je wszystkie do pamięci i zwraca kontrolę do klawiatury. Po wczytaniu linii do pamięci, są one identyczne z liniami programu, które były wpisywane z klawiatury lub odczytane wcześniej z pliku zapisanego na dysku. Z tego względu mogą być używane dowolne zwykłe polecenia takie jak RUN, LIST czy SAVE.

Warto zwrócić także uwagę na to, co zostaje osiągnięte po wpisaniu polecenia SAVE. Jak zawsze, polecenie to po prostu zapisuje program na dysk w standardowym pliku Apple Basic. Jednak program, który pierwotnie znajdował się w pliku tekstowym i został załadowany do pamięci za pomocą polecenia EXEC, zostanie umieszczony z powrotem na dysku w standardowym pliku Basic przy pomocy polecenia SAVE. W ten sposób program może być łatwo przeniesiony z pliku tekstowego do "zwykłego" pliku programu.

Tak samo łatwo można również przenieść program z pliku Apple Basic do pliku tekstowego. Należy go najpierw załadować do pamięci. Dalej może być zapisany na dysku w formie pliku tekstowego, a on z kolei będzie bez problemu wczytany do Magic Window w celu edycji. Aby zapisać program w pliku tekstowym podczas, gdy Applesoft jest uru-



chomiony, trzeba po prostu wprowadzić kod programu w Basicu do pamięci, a na początku dodać trzy poniższe linie:

```
1 D CHR$(4): PRINT D$;
  "OPEN PRG": PRINT D$;
  "WRITE PRG"
2 LIST 4,32767
3 PRINT D$; "CLOSE":
  END: REM
```

Zmienna D\$ oznacza kombinację klawiszy CTRL i D. Następnie, gdy podane jest polecenie RUN, wykonywane są tylko te trzy pierwsze linie, a pozostałe linie programu w Basicu zostaną umieszczone na dysku w pliku tekstowych o nazwie PRG lub dowolnej innej nazwie. Polecenie LIST w wierszu numer 2 powoduje, że wszystkie linie programu o numerach od 4 do 32767 są przenoszone do pliku tekstowego. Do nazwy możemy dodać także rozszerzenie ".MW", które spowoduje, że Magic Window będzie automatycznie rozpoznawał nasz plik jako własny.

Pliki zapisane w Applesoft nie zawierają znacznika End-Of-Text wymaganego przez edytor, dlatego po wczytaniu do Magic Window wystąpił błąd, który jednak można ignorować. Odpowiednie zakończenie pliku zostanie automatycznie wstawione później, gdy Magic Window zapisze plik, więc później już nie pojawi się komunikat o błędzie. Gdy Magic Window załaduje plik i wyświetli go w edytorze, nie zobaczymy także instrukcji RUN.

W tym artykule starałem się opisać, jak edytor tekstu Magic Window może być używany jako pomoc programistyczna. Jako sam edytor, program jest lepszy od interpreterów Applesoft i Integer Basic. Inne edytory tekstu mogą być również odpowiednie do edycji programu, o ile mogą operować na zwykłych plikach tekstowych.

Dobry edytor nie tylko powinien ułatwiać wprowadzenie programu i modyfikacji. Powinien być przydatny

także po to, aby programista mógł zachować proste zasady kompozycji listingi, takie jak struktura akapitów. Może to w znacznym stopniu pomóc w dalszej pracy dzięki wyraźnemu podzieleniu sąsiadujących podprogramów. Jest to szczególnie ważne, jeśli nad programem ma pracować kilka osób.

Korzyści płynące z zastosowania edytora tekstu do tworzenia i rozwijania programów w Basicu są osiągalne kosztem używania plików tekstowych. Piszę "kosztem", ponieważ obsługa takich plików jest wolniejsza i zajmują one więcej miejsca na dysku. Jednak niezwykle łatwo można przenosić programy pomiędzy interpreterem Basica i edytorem tekstu, dlatego wspomniane wady są ważnym czynnikiem właściwie tylko podczas edycji programu.

Przyszła mi do głowy również pomysł, aby wykonać kilka kopii dyskietki z programem Magic Window i na każdej z nich zapisać podstawowe listingi, nad którymi chcemy pracować, o ile będziemy zajmować się różnymi projektami. Dawniej program był zabezpieczony przed kopiowaniem, dzisiaj można stworzyć własne dyskietki "robocze", aby nie trzeba było wymieniać nośników podczas przenoszenia danych.

Na koniec dodam, że w Internecie można znaleźć oryginalną instrukcję do programu Magic Window. Ma tylko 70 stron i opisuje podstawowy sposób korzystania z edytora, ale jest to wystarczające, aby wykonać czynności opisane w artykule.

atari  
online.pl

**Serwis informacyjny  
o komputerach ATARI 24h  
proponujemy:**

- gry i programy użytkowe
- opisy, instrukcje
- historia ATARI, wywiady
- naprawy, rozszerzenia
- programowanie 6502

**a także:**

- organizujemy stoiska,
- majówki
- odczyty

**NAJWIĘKSZY**  

wyślij dyskietkę  
kopertę ze  
znaczkiem

XL/XE
ST/STe

**odeślemy  
katalog**  
**W Y B Ó R**

**SERWIS ATARI**  
**monitory przerobki**  
**telefon**  
**696 030 914**

Magazyn użytkowników komputerów Commodore

**Z pasją o kulturze retro gier komputerowych:**

- recenzje ■ felietony ■ tutoriale ■ dyskusje
- demoscena ■ sztuka ■ programowanie
- wywiady ■ emulacja ■ nowości

Komoda & Amiga plus to pismo poświęcone szerokokopijnej kulturze retro gier komputerowych. Mimo że nasze rozważania poświęcone są głównie produkcjom publikowanym na sprzęt spod znaku marki Commodore to jednak celem nadrzędnym jest odświeżenie tego niesamowitego klimatu lat 80/90-tych, czyli czasów naszego dzieciństwa wypełnionego pikselami i dźwiękiem chipowym.

**team@ka-plus.pl**  
**www.ka-plus.pl**  
 Amiga 68k, AmigaOS 4.x, AROS, MorphOS  
 Plus/4, VIC20, C16, C64, C128

# Poznajemy system RISC OS 3

Opracował:  
**MARIUSZ  
WASILEWSKI**

System RISC OS został pierwotnie wydany w 1987 roku jako wersja Arthur 1.20. Kolejna wersja, określana jako Arthur 2, stała się później znana jako RISC OS 2 i została udostępniona w kwietniu 1989 roku. RISC OS 3.00 został wydany wraz z komputerem A5000 w 1991 roku i zawierał szereg nowych funkcji. Warto dodać, że do 1996 roku system RISC OS został zainstalowany na ponad 500 tys. komputerach.

System RISC OS został pierwotnie wydany w 1987 roku jako wersja Arthur 1.20. Kolejna wersja, określana jako Arthur 2, stała się później znana jako RISC OS 2 i została udostępniona w kwietniu 1989 roku. RISC OS 3.00 został wydany wraz z komputerem A5000 w 1991 roku i zawierał szereg nowych funkcji. Warto dodać, że do 1996 roku system RISC OS został zainstalowany na ponad 500 tys. komputerach.

Trzecia generacja systemu wprowadziła wiele zmian i poprawek w stosunku do RISC OS 2.00. Generalnie można je podzielić na dwa obszary: dodatkowe funkcje dla użytkownika oraz dodatkowe interfejsy programowania dla developerów. Główna zaleta dla użytkownika to fakt, że komputer stał się łatwiejszy w obsłudze, natomiast jeśli chodzi o programowanie, producent oczekiwał wykorzystania

nowych funkcji przez twórców oprogramowania w celu uzyskania bardziej wydajnego oprogramowania.

Wśród ulepszeń w RISC OS 3 są zmiany na pulpicie systemowym. Firma Acorn chciała wyeliminować konieczność używania linii poleceń i zgodnie z tą strategią na pulpicie udostępniono większą ilość funkcji. Wszystkie operacje uruchamiane z pulpitu zaczęły funkcjonować w sposób wielozadaniowy, ponadto pojawił się wskaźnik wolnego miejsca w napędach, który umożliwia ciągły podgląd wolnego miejsca

na dysku. Możliwe stało się wyszukiwanie plików oraz ustawianie typu plików z wygodnego menu.

Dowolny program może być ustawiony tak, aby uruchamiał się automatycznie po włączeniu komputera. Wiele czcionek zostało przeniesionych do pamięci ROM, podobnie jak aplikacje dostępne za pomocą ikony Apps po lewej stronie paska ikon.

Wyeliminowano ograniczenia co do liczby okien, które można otworzyć jednocześnie. Okna mogą być też częściowo wysunięte poza ekran, co



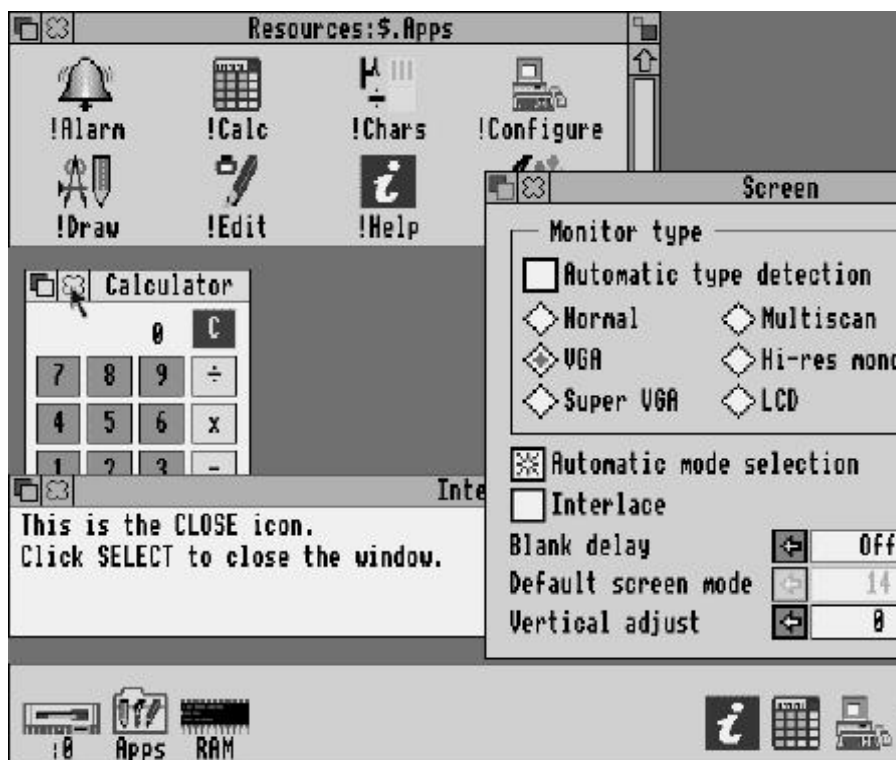


pozwała zwiększyć przestrzeń roboczą użytkownika. Często używane pliki mogą być przechowywane na pulpicie, co ułatwia dostęp. Ważną funkcją jest też opcja zapisania stanu systemu i przywracania go przy następnym uruchomieniu komputera. Otwarte okno można zwinąć do ikony i zapisać na pulpicie. Można je później przywrócić do poprzedniego rozmiaru, po prostu klikając na ikonę.

System umożliwia jednocześnie podłączenie więcej niż jednej drukarki do komputera, przy czym wszystkie podłączone drukarki są dostępne z poziomu pulpitu. Możliwe jest również używanie dysków, które zostały sformatowane na komputerze z systemem MS-DOS. Pozwala to w prosty sposób przenosić dane do RISC OS.

Dyski twarde, którym nadano nazwę mają swoją nazwę wyświetlaną pod ikoną na pasku ikon, co eliminuje potrzebę odwoływania się do tego samego dysku zarówno poprzez podanie jego nazwy, jak i numeru. Ikony tego samego typu są wyświetlane na pasku ikon w grupie, niezależnie od kolejności w jakiej zostały dodane do paska. Na przykład wszystkie ikony dysku twardego będą wyświetlane w grupie, podobnie jak wszystkie ikony serwera plików. Ułatwia to odnalezienie określonej ikony.

RISC OS 3 może obsługiwać do 128 zadań działających jednocześnie, w przeciwieństwie do wersji 2.00, które pozwalała na pracę tylko 32 procesów. Pojawiły się też nowe klawisze funkcyjne służące do otwierania okna, przesuwania paska ikon do przodu i wyłączania programu.



Nie mniej wysiłku włożono w zapewnienie dodatkowych interfejsów programistycznych. Ich ulepszenie miało być inwestycją w przyszłość. Dodajmy, że zachowano zgodność, to znaczy do RISC OS 3 dodano nowy interfejs, ale zachowano również stary - dostępny w systemie w wersji 2.00.

Acorn nie zalecał, aby nowe oprogramowanie napisane dla RISC OS 3 korzystało ze starych interfejsów, ale nie zablokowano możliwości ich używania. Jednak w większości przypadków stare interfejsy szybko okazały się mniej wydajne.

Firma Acorn zaczęła też myśleć o zagranicznych użytkownikach swoich komputerów. Stwierdzono, że wymaga to nie tylko tłumaczenia całego tekstu w systemie na język obcy, ale również dodania możliwości obsługi różnych stref czasowych, alfabetów i układów klawiatur. Drugim aspektem było

umożliwienie pisania oprogramowania w sposób umożliwiający łatwe dostosowanie do rynku zagranicznego. Trzeci aspekt to opcja pisania oprogramowania, które może obsługiwać więcej niż jeden język w tym samym czasie.

Oczywiście system rozwijał się dużo później, ale wersja RISC OS 3 jest moim zdaniem przełomowa. Kolejne wydania działały zaczęły działać na maszynach wirtualnych, jak również urządzeniach typu Raspberry Pi. Ogólny schemat działania oraz układ pulpitu nie został zmieniony, przez co nowe wersje można traktować jako system Nowej Generacji, który zachowuje podstawowe elementy pierwowzoru.

Jeśli interesuje Was bardziej praktyczne podejście do RISC OS, odsyłam do poprzednich numerów naszego magazynu, gdzie pisaliśmy dużo na ten temat.

# Komputery edukacyjne

MARCIN LIBICKI

**BBC Microcomputer System, w skrócie BBC Micro, to seria komputerów i związanych z nimi urządzeń peryferyjnych zaprojektowanych i zbudowanych przez firmę Acorn Computer w latach 80-tych. Główny nacisk został położony na edukację, wytrzymałość, oprogramowanie oraz możliwości rozbudowy. Acorn wykorzystał tę maszynę także do rozwoju architektury ARM, która jak wiemy dopiero wiele lat później odniosła sukces.**

Pod marką BBC zostało wyprodukowanych 9 różnych komputerów, ale wyrażenie "BBC Micro" jest zwykle używane w odniesieniu do pierwszych sześciu modeli, czyli: Model A, B, B+64, B+128, Master 128 i Master Compact. Kolejne modele są uważane już za część serii o nazwie Archimedes.

BBC chciała oprzeć swój projekt na komputerze pozwalającym wykonywać operacje jako podstawa do prezentacji w programach telewizyjnych. Lista tematów obejmowała programowanie, grafikę, dźwięk, muzykę, teletext, sterowanie zewnętrznym sprzętem oraz sztuczną inteligencję. Opracowano ambitną specyfikację komputera, a cały projekt był konsultowany z kilkoma firmami, w tym Acorn Computers i Sinclair Research.

W tym czasie zespół Acorna pracował już nad następcą istniejącego

komputera Atom. Zawierał lepszą grafikę i szybszy procesor MOS 6502 z zegarem 2 MHz. Maszyna była wtedy dopiero na etapie projektowania, a zespół miał tydzień na zbudowanie działającego prototypu. Prace trwały przez całą noc, aby uzyskać pracujący model o nazwie

Proton razem, który mógł być pokazany firmie BBC. Proton nie tylko był jedyną maszyną pasującą do specyfikacji BBC, ale także przewyższał ją w niemal każdym parametrze. Na podstawie prototypu BBC podpisało umowę z firmą Acorn już w lutym 1981 r.



Komputer posiadał 16 kilobajtów pamięci RAM, później rozszerzoną do 32 kilobajtów. Cechą, którą współdzielił z innymi komputerami z procesorem 6502, takimi jak Apple i Commodore był fakt, że pamięć była taktowana dwa razy szybciej niż procesor, z naprzemiennym dostępem do procesora i kontrolera wideo. Aby procesor mógł pracować z pełną prędkością (czyli 2 MHz), system musiał być w stanie wykonać cztery miliony cykli dostępu do pamięci na sekundę.

Jedyną firmą, która w tamtym czasie stworzyła pamięć działającą w ten sposób była Hitachi. Większość konkurencyjnych komputerów działała wolniej (np. Amstrad CPC i ZX Spectrum) lub pamięć wideo była całkowicie oddzielona od puli adresów procesora (np. MSX).

BBS Micro posiadał szereg interfejsów wejścia/wyjścia: porty szeregowy i równoległy, 8-bitowy port wejścia/wyjścia cyfrowego ogólnego przeznaczenia, port oferujący cztery wejścia analogowe, wejście pióra świetlnego oraz złącze rozszerzeń, które umożliwiało podłączenie dodatkowego sprzętu. Można w nim zainstalować dodatkowe pamięci ROM i uzyskać do nich dostęp za pomocą mechanizmu stronicowania pamięci. Jako opcja oferowany był interfejs sieciowy oraz interfejs napędu dyskowego.

Wszystkie płyty główne miały miejsce na dodatkowe układy, na przykład interfejs sieciowy, ale był on rzadko montowany. Acorn opracował także możliwość dodanie drugiego procesora Z80, co umożliwiło uruchamianie oprogramowania dla systemu CP/M.



Wbudowany system operacyjny Acorn MOS zapewnił rozbudowane API do obsługi wszystkich standardowych urządzeń peryferyjnych. Specyficzne funkcje niektórych wersji języka Basic, takie jak grafika wektorowa czy makra, zostały umieszczone w pamięci ROM i są możliwe do używania w każdym programie. BBC BASIC, który jest zapisany osobnym ROM-ie, można także zastąpić dowolnym równoważnym językiem programowania.

Co ciekawe, Acorn zdecydowanie zniechęcał programistów do bezpośredniego dostępu do zmieniających się systemowych i sprzętowych. Oficjalnie stwierdzano, że rekomendowane jest korzystanie z wywołań systemowych. Miało to służyć zachowaniu kompatybilności po migracji na nową architekturę. z tego względu tak pisane programy były najłatwiejsze do przenoszenia na inne platformy w całej serii komputerów Acorn.

Na innym sprzęcie powszechnie używane były polecenia odwołujące się do konkretnych komórek pamięci, takie jak PEEK i POKE. W języku BBS BASIC można było w zamian przekazać parametry do funkcji systemu operacyjnego. Tworzone programy rzeczywiście w dużej mierze były zgodne ze wspomnianym API, za wyjątkiem gier, które podobnie jak na innych komputerach odwoływały się do sprzętu w celu uzyskania większej szybkości. Dlatego zwykle wymagały określonego modelu komputera Acorn.

BBC Micro posiadały wystarczającą ilość gniazd komunikacyjnych, aby można było je łączyć w małą sieć. Dzięki temu wiele szkół i uniwersytetów wykorzystywało maszyny w sieciach, a także powstało wiele sieciowych gier dla wielu graczy. Był to swojego rodzaju ewenement w rodzinie komputerów 8-bitowych, które zwykle nie oferowały podobnych funkcji, a jeśli już to nie były wspierane przez producentów.

# Ćwiczmy refleks i intelekt

ALEKSANDER  
KONDAŁA

Uruchamianie gier w systemie MS-DOS to stały element fana retro. Sam posiadam starego peceta i uruchamiam na nim różne tytuły, na zmianę z trybem DOS oraz pulpitem Windows 98. Każdy lubi trochę inne gry, dlatego chciałbym pokazać kilka przykładów gier zręcznościowych oraz takich, w których trzeba pomyśleć.

## Alien Incident

Jest to ekscentryczna dwuwymiarowa przygodówka science-fiction dla gracza w każdym wieku. Zagadki nie są trudne czy nielogiczne, rozgrywka i interfejs są bardzo proste, a fabuła jest zabawna i bez przemocy.

Grasz rolę młodego Benjamin - faceta, który mieszka z ekscentrycznym naukowcem, wujkiem Alberto. Jednak podczas eksperymentu polegającego na odkryciu kosmicznych robaków, Alberto zostaje porwany przez złych kosmitów. Twoją misją jest uniknięcie złapania i uratowanie wujka, a przy okazji oczywiście musisz uratować świat przed zagrożeniem stwarzanym przez tych najeźdźców.

Jeśli chcecie uruchomić tę grę w systemie Windows, może być problem, ale wszystkie znikają po uruchomieniu DOS Boxa. W moim wypadku jedynym utrudnieniem było to, że przy ładowaniu zapisanej gry

program pytał, czy chcę kontynuować, ale tak naprawdę nie miałem zapisanego żadnego stanu gry. Jednak samo udzielenie odpowiedzi "tak" pozwoli wznowić zapisaną grę bez problemu.

Interfejs gry jest chyba tak prosty, jak to tylko możliwe. Poniżej ekranu gry znajdują się rzędy kwadratów, w których mamy przedmioty, które nosi nasz bohater. Po lewej stronie znajdują się strzałki, które umożliwiają przewijanie przedmiotów, a po prawej stronie - ikona dysku, na którą klikamy, aby zapisać, wczytać stan gry lub przejść do ekranu opcji.

Aby wejść w interakcję z dowolnym elementem lub osobą na ekranie, wystarczy kliknąć prawym przyciskiem myszy. Aby uzyskać dostęp do przedmiotów, również wystarczy kliknąć na nie lewym przyciskiem myszy. Jest naprawdę prosto.

Gorąco polecam tę produkcję dla wszystkich grup wiekowych, niezależnie od tego, czy gracie dużo, czy mało.

## Battlezone

Battlezone to gra akcji połączona ze strategią, rozgrywana z perspektywy pierwszej osoby. Gra zyskała powszechne uznanie zarówno recenzentów, jak i graczy.

Fabuła przedstawia się tak: Wyścig kosmiczny był kłamstwem. Pod koniec lat 50-tych XX wieku w Cieśninie Beringa spadł meteor. W gruzach amerykańscy i radzieccy naukowcy odkryli jednocześnie specjalny bio-metal, który mogliby wykorzystać do produkcji niesamowitych pojazdów i nowych broni.

Szybko zebrali cały materiał, który spadł na Ziemię, a po obu stronach konfliktu pozostało pragnienie osiągnięcia nowych celów. To zapoczątkowało prawdziwy wyścig kosmiczny.



Strefa bitwy pozawala na dowodzenie futurystycznymi pojazdami w walce o bio-metal na sześciu różnych planetach naszego Układu Słonecznego. W grze są dwie kampanie: amerykańska kampania NSDF (Stars and Stripes Missions) i rosyjska kampania CCA (Red Brigade Missions). Obie strony mają swoje unikalne jednostki i broń.

Złom biologiczno-metalowy jest jedynym zasobem w strefie bitwy. Surowiec zebrany przez padlinożerców jest przekształcany przez Recycler w jednostki podstawowe: Fabrykę, która buduje bardziej zaawansowane jednostki, Zbrojownię, która przenosi broń w odległe miejsca oraz Platformę Budowlaną, która jest wykorzystywana do budowy budynków, takich jak wieże strzelnicze i hangary naprawcze.

Jednostki podzielone na ofensywne, obronne i użytkowe, są sterowane za pomocą prostego interfejsu. Używając tylko kilku cyfr na klawiaturze

można łatwo zarządzać wszystkimi aspektami dowodzenia bez względu na to, czy budujesz jednostki, umieszczasz konstrukcje, czy zlecasz atak.

Istnieje ponad 30 unikalnych jednostek, które możesz dowodzić i pilotować, takich jak czołgi, bombowce, artyleria oraz różne rodzaje broni - armaty, rakiety i pociski.

Oprócz kampanii dla jednego gracza, na którą składa się 30 misji, możesz zagrać w liczne misje typu Instant Action lub podjąć znacznie trudniejsze wyzwanie w oficjalnym dodatku o nazwie Red Odyssey, który zawiera dwie nowe kampanie.

Jeśli chcesz zdobyć się na coś więcej, będziesz musiał zmierzyć się z ludzkimi przeciwnikami w bitwach typu multiplayer. Jednocześnie może walczyć do ośmiu graczy na mapie deathmatch lub do czterech graczy na mapach strategicznych. Opcja

gry dla wielu graczy jest bardzo ciekawym doświadczeniem, które nadaje tej grze długowieczności.

## Bermuda Syndrome

Większość gier jest inspirowana poprzednimi produkcjami. Bermuda Syndrome nie kopiuje jednak żadnego z wielkich pomysłów wykorzystanych w grach takich jak Doom czy Command & Conquer. Zamiast tego mamy unikalną grę trochę w stylu Another World (lub Out Of This World), a czasem przypomina także Prince of Persia.

Podobnie jak w przypadku tamtych produkcji gra zaczyna się od teleportacji do innego wymiaru. Zamiast jednak skończyć na jałowym, skalistym świecie, rozbijasz swój samolot w bujnej dżungli wypełnionej dinozaurami, tym samym przypadkowo ratując księżniczkę przed raczej niezbyt ciekawym losem.

Od tego momentu poruszasz się po dżungli, gdzie prawie każdy ekran jest zagadką. Możesz skakać, wspinać się, rozmawiać, używać przedmiotów, strzelać - wszystko, czego potrzebujesz, aby ukończyć rozgrywkę.

Trzeba przyznać, że gra nie wybaczają błędów. Można bardzo szybko zrobić zły ruch lub podjąć złą decyzję i zobaczysz napis Game Over. To jest naprawdę tytuł, w którym musisz działać na zasadzie prób i błędów i choć myślenie logiczne doprowadzi Cię do następnego ekranu to bardzo często można natknąć się na sytuacje, w których jakiś sposób nie działa po prostu dlatego, że programiści nigdy o tym nie pomyśleli.



Nadal jednak nie jest tak źle jak w *Another World*, gdzie częste umieranie bohatera może prowadzić do frustracji. *Bermuda Syndrome* powstał w trochę innych czasach, ale gra jest rozplanowana w podobny sposób, jak robiono to przynajmniej kilka lat wcześniej.

Duża zmiana dotyczy grafiki. Nie ma tu grafiki wektorowej, ale zamiast tego są bardzo kolorowe tła w wysokiej rozdzielczości, czyli 640x480, co było uważane wtedy za "high resoluton". Gra po prostu tworzy bardzo przyjemną, wiarygodną atmosferę. Wolałbym jednak, aby dżunglę ożywiła lepsza muzyka. Podobnie jest z animacją, która jest płynna, chociaż moim zdaniem sekwencje różnych ruchów bohatera mogłyby mieć więcej klatek.

Jednak w pewnych obszarach rozgrywka nie układa się zbyt dobrze. Warto wspomnieć, że interfejs nie jest bardzo dobrze dopracowany. Nawet po godzinie rozgrywki wciąż przypadkowo strzelałem z pistoletu zamiast używać przedmiotu. Również sam świat nie jest tak realny jak ten znany z *Another World*.

Każdy ekran stanowi zagadkę do rozwiązania, ale jest to dziwne uczucie, bo tracisz poczucie, że jesteś naprawdę był w środku ogromnej dżungli. Pomogłyby w tym puste ekrany, w których nie byłoby nic do zrobienia, zamiast wstawiać wszędzie dinozaury i inne przeszkody. Z tego powodu możesz poczuć się tak, jakby każdy ekran był oddzielnym, izolowanym miejscem. Byłoby też dobrze, gdyby ekrany były przewijane zamiast statycznych obrazów w tle.

Pomimo tych wad, *Bermuda Syndrome* jest bardzo zabawną grą - jest kolorowa, pełna wyzwań i może być świetną rozrywką dla każdego, kto lubi bardziej klasyczny *Another World*.

## Buggy Ranger

Kolejna gra to produkcja hiszpańskiego programisty z czasów, gdy popularna była 4-kolorowa grafika CGA. Koncepcja jest dość prosta: przejeżdżasz pojazdem przez miasto, które wygląda jak strefa bitwy. Możesz strzelać, a niektóre przeszkody trzeba po prostu przeskoczyć.

Twój pojazd składa się z dwóch części. Jedną z nich jest samochód na kołach przejeżdżający przez ekran, a drugą jest pistolet rakiety. Naciskając przycisk ognia można dostać się do kontroli, a więc możesz strzelać do "złych". Prędkość z jaką przewijany jest ekran zależy od prędkości pojazdu, który może ją utrzymać aż do momentu, gdy zmienisz kierunek lub w coś uderzy.

Poza grafiką CGA, o której już wspominałem, gra zawiera także kilka dźwięków, o których nie warto wspominać oraz syntezowaną mowę, która zawsze była miłym dodatkiem do gier. Nie jest to po prostu plik dźwiękowy, lecz mowa jest generowana przez program. Osobiście bardzo lubię takie rzeczy.

## Blood Omen: Legacy of Kain

Jest to odległy krewniak serii, która była kontynuowana później. Fabuła

rozgrywa się w krainie Nemezis. Grasz główną postać, czyli Kaina. Kiedy gra się zaczyna, jesteś w tawernie i prosisz o drinka. Barman odmawia, więc decydujesz się opuścić to nieprzyjazne miejsce. Po wyjściu na zewnątrz, jesteś atakowany przez żołnierzy i w końcu zamordowany. Jednak to nie koniec Twojej historii.

Kiedy się budzisz, jesteś w niesamowitej krypcie, ale lokalizacja nie jest jedyną rzeczą, która się zmieniła. Jesteś teraz wampirem i posiadasz dziwny wygląd, który może pomóc utrzymać obcych w bezpiecznej odległości. Tutaj rozpoczyna się twoja misja.

Rozglądasz się, zastanawiasz się, jak i dlaczego jesteś w tym miejscu oraz dlaczego nadal żyjesz? Oczywiście można o tym mówić tylko do pewnego stopnia, ponieważ powoli dociera do Ciebie, że dołączyłeś do szeregów "nieumarłych". Po kilku chwilach krwawej akcji, można odkryć portal, który prowadzi do szeregu filarów.

Pojawia się upiorna postać o imieniu Ariel, która informuje Cię, że każdy filar reprezentuje konkretną osobę. Stan fizyczny każdego filaru reprezentuje stan psychiczny połączonej osoby. Gdy ktoś popada w demencję, słup ulega rozkładowi i rozpadowi.

Aby przełamać klątwę utrzymującą Cię w tym nędznym stanie pomiędzy życiem a śmiercią, musisz przywrócić filary do ich pierwotnego stanu. Zrobisz to zabijając wszystkie osoby związane z filarami i zwracając ich artefakty we właściwe miejsca.

W Blood Omen masz możliwość używania zaklęć, najczęściej spotykanych w jaskiniach. Począwszy od ofensywnych, które powodują wszczęcie implantów przez wroga, aż po bardziej nieszkodliwe i praktyczne, które przykładowo oświetlają pokój lub transportują Cię z powrotem do krypty.

Istnieją również zaklęcia, które mogą przekształcić Cię na przykład w wilka lub nietoperza. Tylko co wilk ma do czynienia z wampirem? Nie wiem, ale zaklęcie jest bardzo ważne, ponieważ daje możliwość eliminowania różnych przeszkód. Możesz też przekształcić się w człowieka - jest to przydatne, gdy nie chcesz przyciągać uwagi.

Do rzucania zaklęć potrzebna jest mana, która jest uzupełniana poprzez nabywanie małych niebieskich kulek. Nie mają one nazwy, ale wykonują pewne działanie polegające na uzupełnianiu energii Kaina.

Gra ma kilka niezwykłych cech. Na przykład deszcz lub woda powoduje ból u wampira. Za to tryb walki jest dość dziwny. Nawet jeśli będziesz grać często, trudno będzie Ci trafić wroga. Dzieje się tak dlatego, że możesz rzucać mieczem tylko w czterech kierunkach. Możesz użyć klawiszy na bloku numerycznym dla większej kontroli, ale jest to trudne, więc zdecydowałem się trzymać się normalnych klawiszy strzałek.

Innym dziwnym zachowaniem jest to, że każda postać może Cię uderzyć i zranić, więc jest to miłe, że wampiry są niewrażliwe na uderzenia. Te zmiany były rzecz jasna potrzebne, ponieważ gra nie miałaby sensu, gdyby Twoja postać była niezwyciężona.

Ogólnie gra nie jest łatwa. Wrogowie są silni, a jeśli zostaniesz ranny, możesz użyć zaklęcia lub użyć krwi więźniów, na których natkniesz się w prawie każdej jaskini. Wampiry nie

gryzą szyi, ale używają poręcznego kija, aby odsączyć krew swoich ofiar. W dalszej części gry natkniesz się na sanktuaria, które leczą, a nawet udzielają błogosławieństw. Główna trudność rozgrywki wynika z faktu, że potwory odradzają się i uzupełniają energię, gdy na przykład wracają z jaskini. Jest to w zasadzie ta sama metoda, którą stosuje słynne Diabło.

Gra jest pełna tajemnic, które można odkryć, włączając ukryte przełączniki. Istnieją również mini-zagadki, które muszą być rozwiązane, zanim będzie można przejść dalej. Niestety, nie ma jednego schematu do naśladowania, trzeba to zrobić w klasyczny sposób - poprzez metodę prób i błędów.

Moim zdaniem najlepszą cechą gry jest atmosfera. Muzyka jest wyjątkowa i dodaje wiele klimatu. Grafika w jaskiniach jest szczegółowa, realistyczna i ma mroczny charakter. Potężna atmosfera wywołana przez muzykę i grafikę naprawdę pozwala doświadczyć dziwnego świata Kaina.

Kolejną cechą, która najbardziej przyczynia się do atmosfery gry są dialogi. Za każdym razem, gdy natkniesz się na coś nowego - nowe zaklęcie lub zwrot w fabule - postać ma coś ciekawego do powiedzenia. Język, którym się posługuje również jest wyjątkowy. Tak naprawdę nie potrafię tego dokładnie wytłumaczyć, ale dialogi są jedne z najlepszych, jakie widziałem we wszystkich grach jakie znam.

## Carnage

Szybkie samochody, palenie opon, materiały wybuchowe i dużo





adrenaliny - wszystkie te elementy są połączone w tej małej grze wyścigowej pod tytułem Carnage. Czy jesteś wystarczająco odważny, aby spróbować szczęścia na torze wyścigowym?

Logo marki samochodu jest narysowane przy użyciu zbyt małej ilości pikseli, aby można było ją łatwo zidentyfikować, ale samochód jest szybki. Tory w grze są pokazywane jako widok z góry. Każdy etap mieści się na jednym ekranie, więc nie są potrzebne żadne efekty przewijania, a Ty jako gracz możesz zobaczyć dokładnie, gdzie jest każdy z Twoich przeciwników. Mogą nie być oni najmądrzejsi, ale bardzo dobrze wiedzą, jak prowadzić samochód. Jeśli jeden z nich uzyska zbyt dużą przewagę, możesz zapomnieć o wygranej. Jest tylko jeden sposób na zakwalifikowanie się do następnego wyścigu: musisz być numerem jeden na torze.

Ogólny poziom trudności nie jest bardzo wysoki, ale cały czas jeden drobny błąd może prowadzić do utraty wyścigu. To może być frustrujące, gdy jesteś w czołówce i nagle wpadasz na kogoś tracąc dobrą pozycję. Niektórych z tych sytuacji można uniknąć, poza tym możesz użyć Nitro jako akceleracji, aby wyprzedzić przeciwników.

W grze jest szesnaście torów, ale muszę przyznać, że nie udało mi się zobaczyć ich wszystkich. Są dwa rodzaje: klasyczne tory i dwukierunkowe drogi z pętlami na końcu. Każdy rodzaj ma swoje plusy i minusy. Jeśli tor jest klasyczny, samochód, który po pierwszym okrążeniu wychodzi na prowadzenie jest naturalnie wytypowany, aby

wygrać wyścig. Może on umieścić miny, aby spowolnić innych kierowców. Jeśli przegapisz szansę na pierwszym okrążeniu - prawdopodobnie nie awansujesz do następnego wyścigu.

Drugi typ toru jest bardziej złożony, ponieważ samochody mogą się udierać w siebie jadąc w różnych



kierunkach. W tym przypadku wszystko może się zdarzyć. Za wszelką cenę należy unikać min i kolizji, ponieważ nie jest potrzebne wiele, aby zmienić kierowców na listę liderów.

Pomiędzy wyścigami Twój samochód może być ulepszony w zależności od Twoich dotychczasowych wygranych. Możesz poprawić sterowność pojazdu, przyspieszenie lub prędkość. Zwiększenie prędkości jest zauważalne, ale jeśli chodzi o ulepszenie układu kierowniczego, nie zauważyłem różnicy w prowadzeniu. W sklepie można również uzupełnić

zapasy min i nitro, ponieważ te elementy są stałe ilościowo na cały turniej. Musisz z nich korzystać mądrze.

Niestety prawie nic nie można ustawić w opcjach gry. Możesz wybrać jednego lub dwóch graczy, klawiaturę lub joystick i zdefiniować klawisze. Ostatnia opcja nie jest za-

pisywana po wyjściu z gry, więc prawdopodobnie będziesz zmieniał klawisze przy każdym rozpoczęciu gry, ponieważ domyślne ustawienia (X i C - kierunki, V do przyspieszenia) nie są zbyt wygodne.

Ogólnie Carnage jest naprawdę przyjemną grą i może zatrzymać gracza przed monitorem o "jeszcze tylko jeden wyścig" dłużej. Grafika jest wyraźna i choć dźwięki podczas wyścigu nie są niczym szczególnym, nie są też irytujące, jak w niektórych innych grach.

# Dzielenie tekstu w IBM Basic

Opracował:  
**KAMIL STOKOWSKI**

Oto prosta sztuczka, która pomoże Ci pisać programy kompatybilne z 40- lub 80-kolumnowym trybem ekranu. Działa ze wszystkimi wersjami IBM Basic.

Komputery IBM mają możliwość korzystania z 40-kolumnowego lub 80-kolumnowego trybu tekstowego. Opcja 40 kolumn jest konieczna, ponieważ na większości telewizorów litery są zbyt małe i niewyraźne, aby były dobrze widoczne w trybie 80 kolumn. Przełączanie tych trybów sprawia nieco problemów programistom. Jeśli program jest napisany w trybie 80 kolumnowym, niektóre słowa, które są wyświetlane na ekranie mogą być podzielone, gdy program będzie uruchomiony w trybie 40 kolumnowym. Jednym ze sposobów uniknięcia tego problemu jest wstawienie wystarczającej ilości znaków Spacji przed słowem, które zostałoby podzielone, tak aby rozpoczynało się już w następnej linii w trybie 40-kolumnowym. Niestety, czasami wygląda to bardzo dziwnie po włączeniu trybu 80-kolumnowego.

Bardziej skutecznym sposobem rozwiązania tego problemu jest wykorzystanie funkcji wbudowanej w IBM BASIC. Gdy instrukcja PRINT drukuje serię ciągu tekstowych odd-

zielonych średnikami, sprawdza czy każdy tekst zmieści się w obszarze pozostałym w bieżącej linii. Jeśli nie, ciąg automatycznie rozpoczyna się od następnego wiersza. Na przykład, jeśli A\$ i B\$ mają długość 30 znaków, linia:

```
PRINT A$;B$
```

wypisze A\$ i B\$ w tej samej linii, jeśli ekran jest w trybie 80 kolumn, ale w różnych liniach na ekranie 40 kolumn. Zmienna B\$ będzie widoczna w następnej linii, ponieważ jej zawartość nie zmieści się w tej samej linii razem z A\$.

Dlatego, dzieląc tekst w ten sposób PRINT możesz być pewien, że słowa nie zostaną podzielone. Wystarczy policzyć znaki, aż do numeru czterdziestego, a następnie cofnąć się aż do Spacji i podzielić tekst na dwie części oddzielone średnikami. Na przykład, zamiast:

```
PRINT "PRESS M FOR  
MENU, R TO REPEAT  
OPERATION"
```

wpisujemy:

```
PRINT "PRESS M FOR  
MENU, R TO REPEAT;";  
"OPERATION"
```

Jeżeli druga część ciągu ma więcej niż 40 znaków, może być konieczne podzielenie tekstu w linii z instrukcją PRINT na kilka części. Korzystanie z ciągów tekstowych jest trudniejsze, jeśli nie możesz z góry przewidzieć, ile znaków będzie miał wprowadzany tekst. Taka sytuacja występuje, gdy chcesz umożliwić użytkownikowi wprowadzanie danych. W takiej sytuacji najprostszym sposobem jest limitowanie ilości znaków w taki sposób, aby zmienna była wyświetlana tylko do określonego znaku.

Można to osiągnąć za pomocą funkcji LEFT\$ podając w nawiasie nazwę zmiennej i ilość znaków do wyświetlenia. Na przykład linia:

```
PRINT LEFT$(A$,10)
```

spowoduje wypisanie pierwszych 10 znaków zmiennej A\$..

# Rozszerzenie GSX do języka CBasic

MARCIN LIBICKI

Digital Research, producent systemu CP/M, w ostatnim okresie działalności rzucił rękawice na rynku komputerów 16-bitowych. Dodał do swojej oferty popularne kompilatory działające w środowisku PC-DOS. Miałem okazję przyjrzeć się dwóm z nich - mam na myśli: Pascal MT+ i CBasic z rozszerzeniem graficznym GSX-86. Oba języki posiadają funkcje niedostępne w porównywalnych wersjach stworzonych przez Microsoft, na przykład możliwe jest bezproblemowe wywoływanie funkcji DOS z poziomu programu.

Kompilator CBasic posiada poręczne zmienne i funkcje, takie jak COMMAND\$, które wycofują łańcuch instrukcji używany podczas działania programu. Kolejne słowo MATCH zwraca pozycję pierwszego pojawienia się określonego wzoru znaku w ciągu tekstowych, SADD zwraca adres określonego ciągu, natomiast UCASE\$ zamienia małe litery na wielkie.

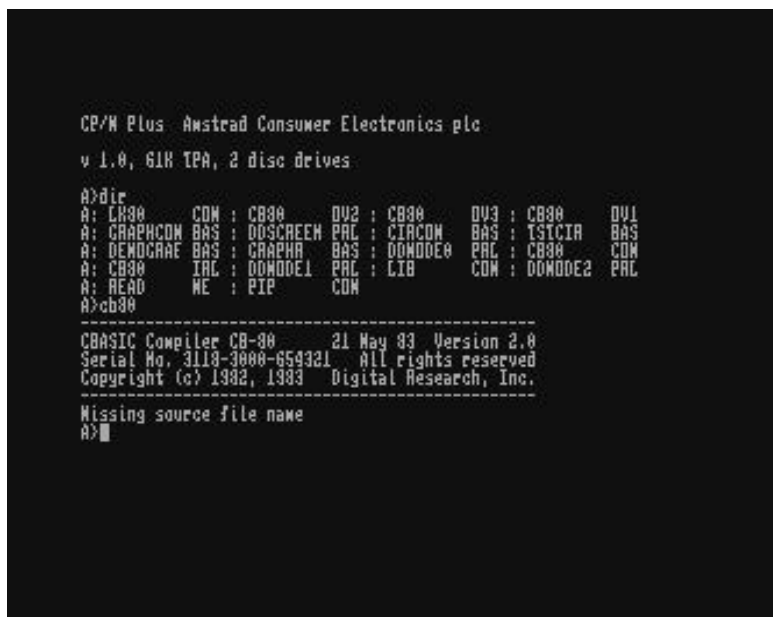
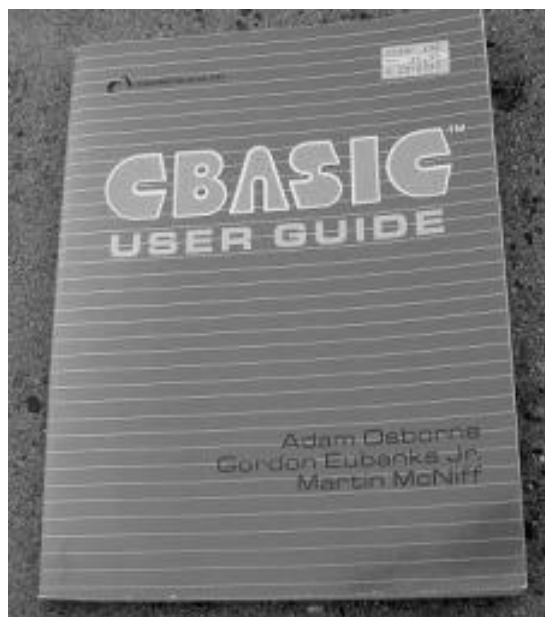
W CBasic istnieje również kilka instrukcji i funkcji przeznaczonych do użycia w środowiskach wieloużytkownikowych lub wielozadaniowych. Przykładowo, ATTACH zwraca wartość logiczną wskazującą, czy określona drukarka jest dostępna do użytku w programie. Jeśli drukarka jest dostępna, ATTACH dołącza ją do programu, umożliwiając w ten sposób dostęp. Słowo DETACH robi coś odwrotnego,

czyniąc drukarkę niedostępną dla programu. Funkcje LOCK i odwrotnie działająca UNLOCK służą do zarządzania dostępem do zapisu w sytuacji pracy wielu użytkowników. LOCK zapobiega modyfikacji przez program, natomiast UNLOCK "odbezpiecza" zapis.

Oryginalna dokumentacja tego języka nie jest dobrze opracowana, ponieważ wydaje się nie być kompletna. Istnieje kilka słów kluczowych, dla których nie podano żadnego wyjaśnienia. Zarówno kompilatory MT+ jak i CBasic mają problemy z obsługą zakończenia pliku kodu źródłowego. MT+ bezwzględnie wymaga dodania znaku końca linii (czyli ENTER) po ostatnim słowie END. Jeśli zostanie to pominięte, kompilator zawiesza się, co wymaga twardego resetu.

Kompilator CBasic był także niezadowolony ze sposobu w jaki mój edytor tekstu kończy plik, ponieważ nie wstawia żadnych znaków kontrolnych. W związku z tym odnalezionych zostało aż 47 błędów "nieprawidłowego" znaku w pliku źródłowym. Tak się stało, bo kompilatorowi nie udało się zlokalizować znaku odpowiadającego kombinacji klawiszy CONTROL + Z, także po poleceniu END. Za to, tym razem program nie wziął pod uwagę, czy nacisnąłem na końcu Enter, czy też nie.

Oryginalny CBasic, działający pod kontrolą CP/M, nie posiadał obsługi funkcji graficznych dla IBM PC. Na szczęście ludzie z Digital Research naprawili to przeoczenie, gdy ponownie wydano wersję dla PC-DOS. Rozszerzenie graficzne CBasic jest dość kompletne i rozbudowane



na pierwszy rzut oka, ale w oryginalnej instrukcji nie opisano wszystkich funkcji.

GSX-86 jest zaprojektowany tak, aby był niezależny, to znaczy, że wszystkie graficzne urządzenia wyjściowe są dostępne także w pozostałej części programu, bez potrzeby modyfikacji. Jeśli na przykład zdecydujesz się na wydruk efektów pracy, zamiast wyświetlić je w konsoli, nie są wymagane żadne zmiany w kodzie programu.

W module graficznym znajduje się około 25 poleceń, które wzbogacają CBASIC. Niektóre z nich mają odpowiednik w Microsoft Basic, na przykład CLEAR, CLS, MAT FILL i PAI. Niestety, w module GSX-86 nie ma żadnych instrukcji porównywalnych z PSET, nie ma też żadnego sposobu na ustawienie koloru tła wyświetlacza.

Pozostałe wyrażenia CBASIC są nietypowe i są bardziej zorientowane na grafikę typu prezentacje danych

niż programy do rysowania. Kilka instrukcji jest przydatnych podczas tworzenia klasycznych wykresów. Słowo kluczowe CHARACTER HEIGHT określa wysokość liter w stosunku do długości współrzędnej pionowej, TEXT ANGLE ustawia pochylenie tekstu wyliczane względem poziomu, a LINE STYLE umożliwia ustawienie różnego rodzaju linii z przeznaczeniem do użycia z poleceniem PLOT lub MAT PLOT.

Dostępne są cztery standardowe style: ciągłe, przerywane, wykropkowane oraz połączenie dwóch ostatnich. Słowo ASK STYLE COUNT określa styl linii dostępny na bieżącym urządzeniu wyjściowym. Można także definiować tak zwane markety, które są specjalnymi symbolami przeznaczone do podświetlenia punktów, linii i wykresów. Znaczniki są definiowane poprzez słowo MARKER TYPE, które opisuje typy znaczników. Kolejne - SET MARKER HEIGHT - określa wysokość znaczników względem współrzędnej pionowej. Współrzędne X i Y może

być rysowana także za pomocą markerów ze statusem MAT MARKER, co jest bardzo przydatne przy rysowaniu wykresów.

Zakres układu współrzędnych jest ustalany za pomocą SET WINDOW, po którym następują granice współrzędnych X i Y. Polecenie to może być również używane do zmiany proporcji grafiki. Rzeczywiste granice mogą być zmieniane za pomocą SET VIEWPORT, po którym należy podać lewy górny róg, a później prawy dolny róg okna.

Moduł GSX-86 był dostarczany z kodem źródłowym w formie kilku programów demonstracyjnych, a także wiele innych przykładów było podawanych w dokumentacji. Dema nie były zbyt dobre nawet jak na czas, w którym powstały, szczególnie w porównaniu z niektórymi plikami dostarczonymi z dyskami PC-DOS, ale dzięki nim można się zorientować, jakie możliwości ma rozszerzenie graficzne.

# Zastępowanie czcionki Topaz

ADAM ZALEPA

Po instalacji polskiej wersji systemu operacyjnego może się okazać, że w oknach komunikaty - choć już pisane po polsku - nie będą posiadały polskich znaków. Nie ma problemu, gdy chcemy używać Workbenchu z twardego dysku, bowiem wystarczy zainstalować pełny pakiet polskiej lokalizacji. Sytuacja wygląda inaczej, gdy mamy do dyspozycji tylko dyskietki lub emulator Gotek. Musimy wtedy wiedzieć, jak wykonać taką dyskietkę ratunkową oraz jakie pliki "wydobyć" z pakietu lokalizacyjnego. W praktyce trzeba tylko skorzystać z odpowiedniego programu.

Nosi on nazwę „FontPLPatch” i wchodzi w skład pakietu „WFMH LocalePL”, dzięki któremu możemy uzyskać kompletne polskojęzyczne środowisko pracy. Program można uruchomić na kilka sposobów, ale najlepszym jest umieszczenie go dokładnie w tym samym katalogu, co na twardym dysku, czyli „WBStartup”.

Dlatego na dyskietce ratunkowej musisz utworzyć katalog o takiej samej nazwie. Możesz odczytać zawartość nośnika na Workbenchu, a potem wywołać opcję „Nowy katalog” z menu górnego „Okna”. Następnie przenieś ikonę „FontPLPatch” z katalogu „WBStartup” na dyskietkę.

Drugim sposobem jest wykorzystanie okna „Shell” i wpisanie następujących dwóch linii poleceń:

```
makedir Ratunek:WBStartup
```

```
copy SYS:WBStartup/FontPLPatch##? Ratunek:WBStartup
```

albo też, odnosząc się do wbudowanej stacji dyskietek:

```
makedir DF0:WBStartup
```

```
copy SYS:WBStartup/FontPLPatch##? DF0:WBStartup
```

Zauważ, że cały czas korzystamy wyłącznie z podstawowych składników Workbenchu oraz poleceń AmigaDOS, a mimo wszystko uzyskujemy wysoką elastyczność. Gdy teraz zresetujesz Amigę,

Workbench będzie wczytywał się nieco dłużej, bowiem uruchomiony będzie dodatkowy program z katalogu „WBStartup”. Zostanie to wyko-

nane samoczynnie, bez potrzeby zmiany pliku sekwencji startowej.

Gdy stacja dyskietek przestanie pracować, wywołaj preferencje „Input” i sprawdź ponownie klawiaturę w polu „Keyboard Test”. Powinieneś bez problemu uzyskać polskie znaki diakrytyczne, mimo iż reszta programu wciąż komunikuje się w języku angielskim. W naszym przypadku wygląda to tak jak na poniższej ilustracji:

Od tej pory możesz korzystać z polskich znaków we wszystkich systemowych oknach, niezależnie od tego, czy będą to elementy AmigaDOS, Workbenchu czy zewnętrznych programów. W taki sposób poradzisz sobie, gdy z dyskietki ratunkowej chcesz wczytywać całego Workbenchu – wzorem twardego dysku. Jednak ze względu na przykład na

oszczędność pamięci albo długi czas wczytywania, możesz mieć potrzebę uzyskania podobnego rezultatu w oknie AmigaDOS, które będzie widoczne podczas rozruchu dyskiety.

W podobny sposób możesz uzyskać zmianę standardowego kroju czcionek na dowolny inny, który odpowiada rozmiarowi i rodzajowi czcionce „Topaz”. Należy użyć programu o nazwie „Replace Topaz”, który znajdziesz na Aminecie. Przejdź do katalogu „text/font” i pobierz plik o nazwie „ReplaceTopaz.lha”. Archiwum rozpakuj w zwykły sposób, a więc w oknie „Wykonaj polecenie” uzyskaj wpis:

#### **lha x ReplaceTopaz.lha RAM:**

aby zawartość zapisać w „Ram Dysku”. Odczytaj go na Workbenchu, a następnie włącz funkcję wyświetlania wszystkich pozycji w katalogu. Dalej najedź wskaźnikiem na ikonę podpisaną tak jak nazwa programu i naciśnij dwukrotnie lewy klawisz myszki. Pojawią się kolejne ikony, a wśród nich będzie znowu pozycja „ReplaceTopaz”. Tym razem będzie to jednak plik, a nie katalogu, dlatego wykonaj na niej dwuklik. Na ekranie ponownie pojawi się okno „Wykonaj polecenie”, w którym wpisz linię:

#### **copy ReplaceTopaz C:**

i naciśnij ENTER. Jeżeli nie zobaczysz żadnych dodatkowych komunikatów, oznacza to, że operacja przebiegła pomyślnie. Od tej chwili możesz korzystać z nowego programu. Jego sposób obsługi jest bardzo prosty i polega na wpisaniu nazwy czcionki, którą chcesz za-

stąpić oryginalny krój „Topaz”. Na przykład, aby uzyskać ten sam efekt, co za pomocą programu „FontPLPatch”, w oknie „Shell” wpisz:

#### **replacetopaz TopazPL.8**

co oznacza wykorzystanie czcionki „TopazPL” o wielkości 8 punktów. Pomiędzy nazwą a oznaczeniem rozmiaru musisz umieścić kropkę. Po wykonaniu powyższej linii, programy, które używają zazwyczaj ze standardowej czcionki i jednocześnie korzystają z funkcji systemu operacyjnego wczytają inny krój.

Widać to bardzo dobrze, gdy uruchomisz programy w systemowego katalogu „Prefs”. Przykładowo, zmień czcionkę na „XENPL”, który wchodzi w skład pakietu „WFMH LocalePL”. W tym celu w oknie „Shell” musisz wpisać:

#### **replacetopaz XENPL.8**

i naciśnąć klawisz ENTER. Różnica jest zasadnicza.

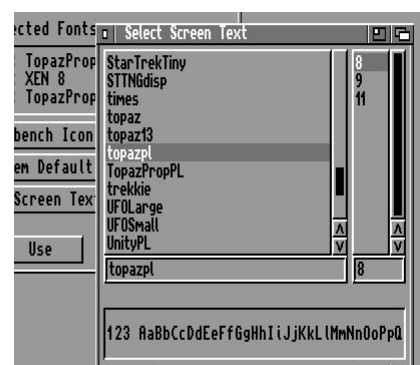
W ten sposób możesz ujednolicić wygląd programów systemowych, nawet tych, które nie dają możliwości zmiany wyglądu. Pamiętaj jednak, że jest to sposób nieprzewidziany przez autorów systemu operacyjnego i niektóre informacje w oknach mogą być wyświetlane nieprawidłowo. Wszystko będzie zależało od konkretnych programów, które uruchamiasz. Ewentualne problemy z wyświetlaniem tekstu nie powinny mieć także wpływu na wywoływane funkcje.

Jeżeli podasz nazwę czcionki, która nie jest aktualnie dostępna w Twoim systemie, w systemie nie zostaną

wykonane zmiany, a jedynie zobaczysz komunikat o treści:

#### **Unable to find font.**

Pamiętaj, że nie możesz zupełnie dowolnie zmieniać krojów, na przykład tak jak w oknach korzystających z pakietu „Magic User Interface”, ale muszą mieć wielkość 8 lub 9 punktów oraz każdy znak musi zajmować tyle samo miejsca na ekranie, czyli nie mogą być to czcionki proporcjonalne.



Mimo tych zastrzeżeń, nie musisz bardzo przejmować się podczas uruchamiania programu „Replace Topaz”, bowiem będzie on sprawdzał, czy wskazałeś odpowiedni krój. Gdy podasz niewłaściwy, żadna operacja podmi- any nie zostanie wykonana. W związku z tym możesz testować różne ustawienia.

Zwróć tylko uwagę na to, że w razie błędnej linii polecenia, zachowana zostanie aktualna konfiguracja. Funkcje systemowe nie powracają samoczynnie do fabrycznych, ani do tych, które zostały wczytane z dysku. Dlatego w niektórych przypadkach, aby wczytać na nowo ustawienia, będziesz zmuszony zresetować Amigę i wykonać ponowny rozruch Workbenchu.



# Pliki ADF

## Tworzenie i kompresja

ADAM ZALEPA

Każdy zna popularny format obrazu amigowych dyskietek o nazwie ADF. Używamy ich w emulatorach, urządzeniach FPGA, jak również prawdziwym sprzęcie. Kiedyś najczęściej używanym formatem był DMS, później zmieniło się to na korzyść ADF, który zazwyczaj się rozpowszechniały w formie nieskompresowanej. Mniej osób wie, że można to zmienić, jak również zna sposoby tworzenia takich plików. Wszystko można zrobić za pomocą różnych programów - ja używam „TransADF”.

Jak to zrobić? Utworzenie tzw. „obrazu” polega na odczytaniu wszystkich sektorów dyskietki i zapisaniu tych informacji w pliku na innym dysku. Aby to zrobić należy posłużyć się następującą formułą:

```
transadf DRIVE=DF0: FILE=Worek:dyskietka1.adf
```

W rezultacie na dysku „Worek” zapisany zostanie plik o nazwie „dyskietka1.adf” zawierający dane uzyskane z dyskietki włożonej do wewnętrznej stacji Amigi. W trakcie wykonywania operacji w oknie będą pojawiać się napisy świadczące o odczytywaniu kolejnych sektorów.

Zwróć szczególną uwagę na linię z napisem „Reading”, która wskazuje na to, która ścieżka jest aktualnie odczytywana. Pamiętaj, że program nie odczytuje poszczególnych plików, nie sprawdza także systemu

plikowego nośnika, lecz tworzy jeden spójny plik archiwum zwany „obrazem”. Dzięki temu możliwe jest odczytanie dowolnej dyskietki, nawet takiej, która określana jest mianem „nie-DOSowej” (ang. Not DOS disk). Mówiliśmy o tym wcześniej, teraz tylko przypomnijmy, że polega to na zapisaniu danych za pomocą nietypowych, zwykle własnoręcznie napisanych mechanizmów, które omijają funkcje systemu operacyjnego.

Standardowo plik „.adf” zajmuje zawsze tę samą objętość 901120 bajtów (czyli 880 kilobajtów), bowiem dane nie są kompresowane, a jedynie zachowywane w takiej samej formie jak bezpośrednio po odczytaniu. Możesz jednak wywołać funkcję pakowania i to na kilka sposobów. Do dyspozycji pozostają trzy opcje nazwane GZIP, ZLIB oraz PKZIP. Oznaczenia te pochodzą od programów

wykorzystujących określone mechanizmy kompresji.

Aby ich użyć wystarczy dopisać jeden z tych wyrazów jako nowy argument na końcu linii polecenia, na przykład

```
transadf DRIVE=DF0: FILE=Worek:dyskietka1.adf PKZIP
```

Zwróć uwagę, że nie zmienia się rozszerzenie pliku, będzie on jednak skompresowany przy wykorzystaniu wskazanej metody. Nie ma to wpływu na dalszą obsługę, bowiem „TransADF” potrafi automatycznie rozpoznać i rozpakować tak utworzone archiwum. Czynność kompresji jest oznaczana osobno, dlatego w oknie zobaczysz dodatkową linię rozpoczynającą się od poniższych słów:

Compressing disk

obok umieszczony będzie symbol metody jaka została wybrana do spakowania danych. Poniżej, tam gdzie normalnie wyświetlane są numery ścieżek, będzie pojawiał się dodatkowy wyraz „Deflating” przeplatany ze standardowym „Reading”.

Cała operacja będzie trwała dłużej, a jej konkretna szybkość zależy od procesora zamontowanego w Twojej Amidze. Na sam koniec wypisana zostanie linia zawierająca dwie wartości, a jej cechą charakterystyczną będzie wyraz „Deflated” umieszczony na początku. Całość będzie podobna do wpisu:

**Deflated: 878592 => 421357 (47%)**

Liczby te wyrażone są w bajtach i określają objętość danych odczytanych z dyskietki oraz rozmiar archiwum po spakowaniu. W naszym przypadku udało się uzyskać wysoki zysk z kompresji, który wynosi 47 procent. Pozostałe opcje związane z kompresją, tj. argumenty GZIP i ZLIB należy używać w taki sam sposób, czyli – odnosząc je do naszego przykładu – tak jak poniżej:

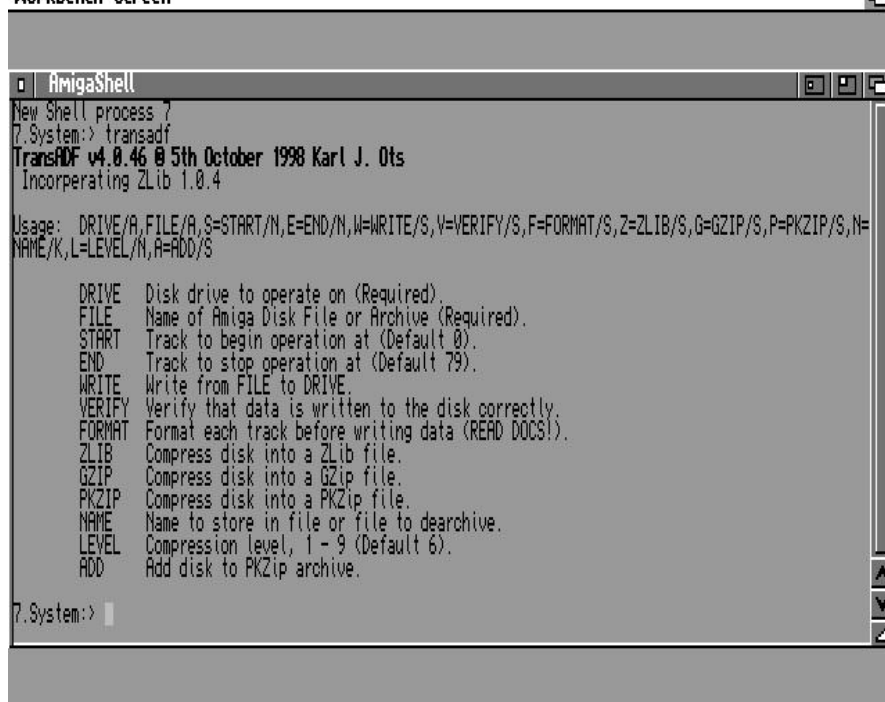
**transadf DRIVE=DF0: FILE=Worek:dyskietka1.adf GZIP**

lub

**transadf DRIVE=DF0: FILE=Worek:dyskietka1.adf ZLIB**

Pisownia znaków nie ma znaczenia, ale warto zachować rozróżnienie między nazwami funkcji a pozostałą częścią linii. Pozwala to uzyskać bardziej czytelne wpisy, a tym samym szybciej zapamiętać sposób użycia programu. Każda z opcji ozna-

#### Workbench Screen



nacza inną metodę pakowania danych. Wybór najlepszej metody zależy od zawartości dyskietki, dlatego nie da się przewidzieć z góry, która funkcja uzyska największy zysk objętości.

Możemy jednak zmieniać stopień kompresji, który przyjmuje wartości w zakresie od 1 do 9. Domyślnym ustawieniem jest 6, lecz możesz to zmienić po podaniu funkcji o nazwie LEVEL. Na przykład:

**transadf DRIVE=DF0: FILE=Worek:dyskietka1.adf PKZIP LEVEL 9**

oznacza użycie algorytmu PKZIP oraz maksymalnego stopnia kompresji, co powinno dać w rezultacie najmniejszy plik „.adf”. Weź pod uwagę, że może to jednocześnie spowodować znaczące zmniejszenie szybkości wykonania całej operacji. Jest to normalny objaw i należy poczekać na zakończenie pracy programu.

Dla „obrazów” dyskietek spakowanych za pomocą opcji GZIP niektóre programy stosują oddzielne rozszerzenie „.adz”. Nie zmienia to jednak sposobu obsługi.

Możliwość kompresji jest bardzo często niedoceniana przez użytkowników dysponujących dużymi dyskami lub przechowującymi swoje pliki na kartach pamięci albo płytach CD i DVD. Warto jednak o tym pamiętać, bowiem praktyka pokazuje, iż ze źródłowych 880 kilobajtów może pozostać rozmiar o ok. 40-50% mniejszy.

Ma to również szczególne znaczenie, gdy chcemy przenosić dane za pomocą stacji dyskietek w formacie MS-DOS. Zmieścimy na niej jedynie 720 kilobajtów i w takiej sytuacji najszybszym rozwiązaniem jest właśnie kompresja. Z drugiej strony trzeba wziąć pod uwagę, że nie wszystkie programy mogą obsługiwać spakowane pliki „.ADF”, dlatego

jeśli pracujesz na większej ilości danych proponujemy najpierw sprawdzić obsługę zarówno odczytu, jak i zapisu dyskietek.

Program „TransADF” umożliwia wykonywanie operacji nie tylko na całych dyskietkach, ale także wybiórczo na określonych ścieżkach. Możemy wskazać początkowy i końcowy numer za pomocą argumentów START i END. Na przykład, aby odczytać ścieżki od 10 do 20 wpisujemy:

```
transadf DRIVE=DF0: FILE=Mo-  
rek:dyskietka1.adf START 10 END  
20
```

Wszystkie inne funkcje działają bez zmian, możemy więc kompresować zawartość i korzystać z weryfikacji zapisu danych. Należy tylko pamiętać o zachowaniu tych samych wartości przy zapisywaniu nowej dyskietki, w przeciwnym razie nośnik nie będzie kopią poprzedniego lub uzyskamy błąd o treści:

#### Error reading

oraz

#### File too short

Gdy chcesz zapisać te same ścieżki na innej dyskietce, po prostu dodaj argument WRITE do poprzedniej linii, czyli razem wpisz następujące polecenie:

```
transadf DRIVE=DF0: FILE=Mo-  
rek:dyskietka1.adf START 10 END  
20 WRITE
```

Co ciekawe, program nie ogranicza użytkownika w modyfikacji numerów. Oznacza to, że po odczytaniu określonych ścieżek możemy je zapisać w całkiem innym miejscu

dyskietki. Musimy tylko zachować taką samą ilość ścieżek. Odnosząc to do naszego przykładu, może to wyglądać na przykład tak:

```
transadf DRIVE=DF0: FILE=Mo-  
rek:dyskietka1.adf START 47 END  
57 WRITE
```

Nic nie stoi na przeszkodzie, aby zapisać tylko część informacji umieszczonych w pliku „ADF”. Należy tylko podać mniejszą ilość ścieżek:

```
transadf DRIVE=DF0: FILE=Mo-  
rek:dyskietka1.adf START 47 END  
52 WRITE
```

Taki sposób użycia programu jest dozwolony i nie wywoła żadnego błędu. Wszystkie opcje razem pozwalają na przeprowadzanie ciekawych eksperymentów, choć oczywiście wiele z nich może nie mieć sensu z punktu widzenia uruchamiania programów. Zwróć jednak uwagę jak bardzo elastycznym programem jest „TransADF”, mimo że jego bardziej zaawansowane funkcje wykorzystywane są raczej rzadko.

Bardzo ciekawą opcją jest możliwość automatycznej archiwizacji dyskietki w pliku z rozszerzeniem „.zip” lub „.gz”. Operacja polega na tym, że po odczytaniu danych i utworzeniu pliku „ADF”, zostanie on dodany do nowego archiwum. Dzięki temu możemy nie tylko uzyskać dodatkową kompresję, ale także umieszczać wiele obrazów dyskietek w jednym archiwum.

Do tego celu nie musimy używać żadnych innych programów poza „TransADF”. Oba wymienione for-

maty bez problemu odczytamy praktycznie na każdym komputerze, dlatego warto nauczyć się z nich korzystać. Pierwszy jest standardowo używany na systemach operacyjnych z rodziny Windows, drugiego cechuje raczej Linux, bowiem należy do kategorii tzw. wolnego oprogramowania.

Obsługa archiwizacji plików „.adf” jest trochę bardziej skomplikowana, lecz znakomicie rozszerza możliwości programu. Na początek musimy użyć dodatkowego argumentu o nazwie NAME, po którym podajemy nazwę obrazu dyskietki. Jednocześnie po słowie FILE trzeba wpisać nazwę archiwum, co stanowi różnicę w stosunku do poprzednich opcji. Łatwo o pomyłkę, dlatego przy korzystaniu z tej funkcji zwróć szczególną uwagę, aby wprowadzić właściwe dane. Na przykład:

```
transadf DRIVE=DF0: FILE=mo-  
je.zip NAME=dysk1.adf
```

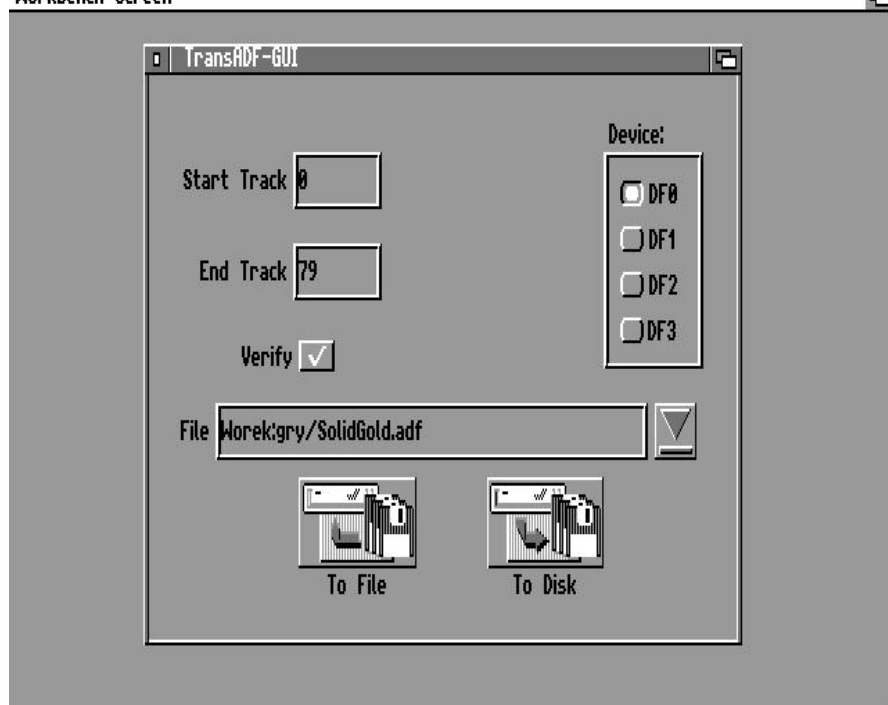
Powyższa linia nie będzie jeszcze działać, gdyż nie wpisaliśmy nazwy mechanizmu kompresji, który zostanie użyty. Służą do tego słowa PKZIP oraz GZIP, które odpowiadają odpowiednio - formatom plików z rozszerzeniem „.zip” i „.gz”. Cała linia polecenia powinna mieć zatem postać:

```
transadf DRIVE=DF0: FILE=mo-  
je.zip NAME=dysk1.adf PKZIP
```

albo

```
transadf DRIVE=DF0: FILE=mo-  
je.gz NAME=dysk1.adf GZIP
```

Nie podaliśmy ścieżek dostępu, przez co pliki zostaną zapisane w



sprawdzić, czy ilość wolnego miejsca na dysku jest wystarczająca do zapisania nowych informacji. W przeciwnym razie operacja nie powiedzie się, a archiwum może ulec uszkodzeniu.

Nie ma sztywnych ograniczeń związanych z ilością plików możliwych do umieszczenia w archiwum „.zip” lub „.gz”. Teoretycznie może być ich nawet kilka albo kilkadziesiąt tysięcy, bo i takie pliki można spotkać w Internecie. W dużej mierze zależy to od szybkości komputera, na którym pliki poddawane są kompresji. Może zdarzyć się jednak, że podasz dwukrotnie tę samą nazwę pliku „ADF”. W takiej sytuacji zostanie on zastąpiony, a program nie spyta o potwierdzenie operacji nadpisania danych. Dlatego proponujemy korzystać przede wszystkim z możliwości dodawania nowych plików do istniejącego archiwum, aby nie spowodować problemów.

katalogu bieżącym, możesz jednak dodać nazwy dysków i katalogów, tak jak wcześniej, aby umieścić archiwa w innych miejscach. Podczas pracy program wyświetli nowy komunikat:

#### Compressing disk as PKZip file lub

#### Compressing disk as GZip file

Funkcje te w niczym nie ograniczają stosowanie pozostałych argumentów, na przykład służących do odczytywania fragmentów dyskietki. Są to dodatkowe możliwości i nie zastępują innych opcji. W jednym pliku „.zip” lub „.gz” możemy też umieścić wiele obrazów dyskietek. Aby tak się stało należy wywołać ponownie tę samą linię polecenia, wprowadzić taką samą nazwę archiwum, lecz inną nazwę pliku „ADF”. Gdy wywołasz poniższy wpis:

```
transadf DRIVE=DF0: FILE=mo-  
je.zip NAME=dysk1.adf PKZIP
```

a następnie drugi:

```
transadf DRIVE=DF0: FILE=mo-  
je.zip NAME=dysk2.adf PKZIP ADD
```

spowodujesz, że do archiwum „moje.zip” dodane zostaną dwa pliki o nazwach „dysk1.adf” oraz „dysk2.adf”. Jest to bardzo przydatne w sytuacji, gdy zapisujemy kilka dyskietek przeznaczonych do przeniesienia do jednego komputera lub mamy do czynienia z wieloma nośnikami związanymi z jednym programem albo grą. Korzystanie z takiego sposobu kompresji niesie za sobą jeszcze jeden skutek w postaci wyświetlenia innego komunikatu, który zawierał będzie linię:

#### Updating archive

Oznacza on, że archiwum zostało utworzone wcześniej, a teraz „TransADF” dodaje jedynie do niego nowe pliki. W tym przypadku przed wywołaniem polecenia należy

atari  
fan magazyn  
wszystkich  
fanów  
Atari

ATARI  
ONLINE.PL

JAGNES

# Gry przełomu roku 1990

ALEKSANDER  
KONDAŁA

W tym numerze naszego magazynu rozpoczynamy dział przeznaczony dla produktów firmy Nintendo. Na początek pokaże kilka gier, które zasługują na uwagę każdego gracza. Nie muszą być to pozycje najbardziej znane. Dla mnie najważniejsza jest grywalność i sterowanie, a dopiero później fajerwerki graficzne.

## Bomberman 2

Jest to sequel słynnej gry, który ukazał się zaledwie cztery lata po swoim poprzedniku. Pod względem technicznym gra jest znacznie lepsza przy jednoczesnym zachowaniu szybkich ruchów, z których znana jest ta seria.

Pojęcie "seria" jest trochę mało precyzyjne, ponieważ tutaj po raz pierwszy w Bombermanie pojawiła się jakakolwiek fabuła. Okazuje się, że nasz bohater ma rywala, który wysłał go do więzienia za domniemaną kradzież banków. Ważne jest to, że grafika przeszła poważne zmiany, a sam Bomberman jest bardziej szczegółowo narysowany.

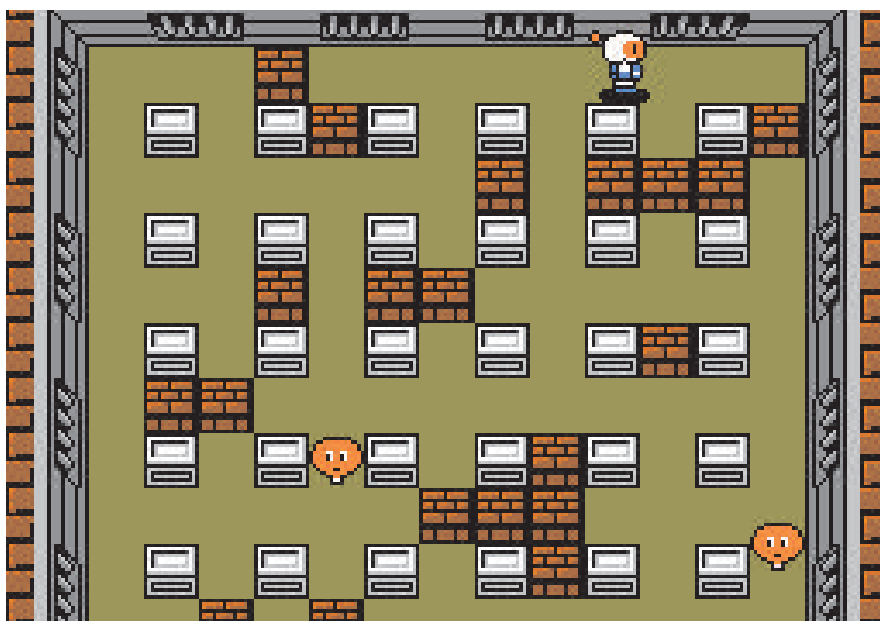
Poziomy podzielone są na sześć części po osiem etapów, z których każdy przedstawia konkretny etap podróży Bombermana przez więzienie, wulkan, góry i las. Musiało to być nie lada gratką dla graczy dawniej. Mimo tych zmian, wrogowie

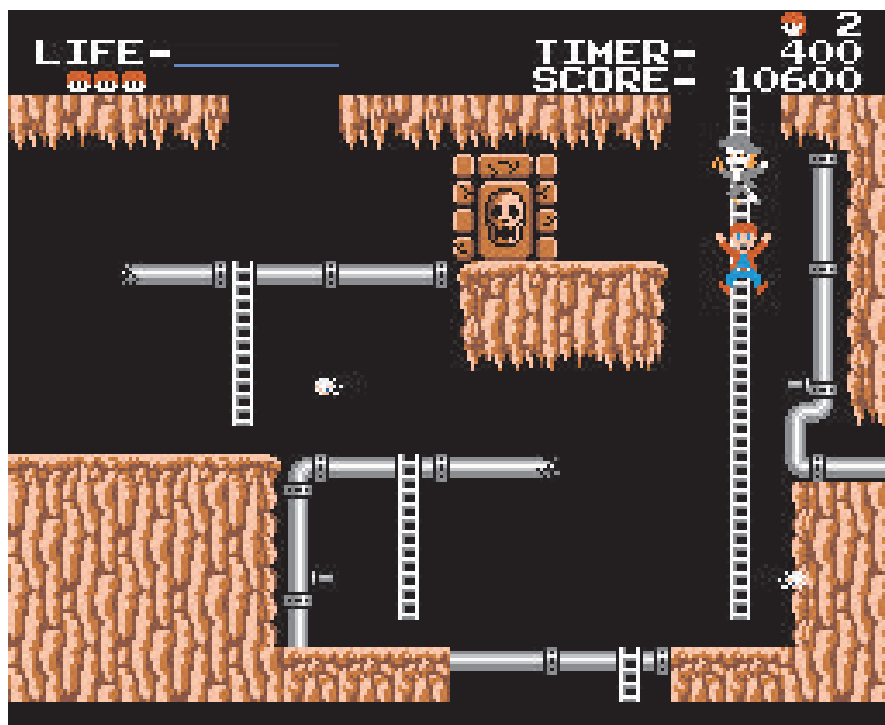
oraz ich taktyka nie zmieniła się w najmniejszym szczególe.

Poziomy są początkowo łatwiejsze, ale później można spotkać dodatkowych "obcych", którzy robią wszystko, co w ich mocy, aby zbić Cię z tropu. Oprócz typowych elementów pojawia się szereg nowych potworów. Jest wróg przypominający smoka, który może przechodzić

przez ściany oraz poruszający się wystarczająco szybko.

Mechanika Bombermana 2 nie zmieniła w porównaniu z pierwszą grą, co jest dobrym znakiem, że wszystko działa tak jak powinno. Dodatkowe życia nie są ograniczone, ale wchodzi w skład nowej linii bonusów. Bomberman '93 na PC Engine używa tej samej mechaniki i





to działa dobrze, ale w sytuacji, gdy zabraknie Ci życia, dostajesz hasło do kontynuacji, więc gra się bardzo przyjemnie.

Wśród dostępnych bonusów Bombermana jest kamizelka ognioodporna, ale działa tylko przez około 30 sekund. Niestety trzeba się domyślić, ponieważ jedynym wskaźnikiem na ekranie jest Bomberman migający do momentu, gdy sygnał dźwiękowy ostrzega, że wkrótce bonus wygaśnie. Wydaje mi się, że to trochę mylące.

Timer daje mniej czasu na ukończenie etapów w miarę przechodzenia do kolejnych poziomów, a gdy skończy się czas Bomberman nie dostaje żadnego dodatkowego czasu - po prostu wrogowie go gonią.

Jeśli chodzi o muzykę, mamy to samo co na PC Engine. Co więcej, cztery nowe motywy są dość krótkie, ale nie są męczące.

Mechanika Bombermana z pewnością może nie zadowoli graczy powracających do tematu retro i poszukujących czegoś nowego. Jednak tryb multiplayer jest dużym krokiem naprzód. Dla mnie jest to jedna z najlepszych produkcji.

## Goonies

Goonies został zaadaptowany przez film Stevena Spielberga z lat 80-tych. Dziś można użyć emulacji, ale dalej najlepszym sposobem jest używanie prawdziwej konsoli. Gra podąża luźno za głównym motywem filmu, tyle że tutaj trzeba ratować inne dzieci, które są uwięzione w różnych podziemnych lochach i jaskiniach pod restauracją, która jest punktem wyjścia.

Aby zagłębić się głęboko w te jaskinie, musisz zebrać kilka kluczy oraz uratować jednego z kolegów uwięzionych za drzwiami. Za każdym razem, gdy grasz, losuje się lokaliza-

cja przedmiotów i zakładników, dzięki czemu gra staje się w pewnym stopniu nieprzewidywalna.

Jedynym sposobem na otwarcie drzwi jest wysadzenie ich z bombami. Musisz być również ostrożny, aby nie zostać uszkodzonym przez wybuch, w przeciwnym razie stracisz od razu całe życie.

Wśród innych przedmiotów zbierasz również klejnoty, które mogą automatycznie uzupełnić Twoją energię, jeśli masz ich wystarczająco dużo. Masz też eliksiry zdrowia, a także procę z ograniczoną ilością amunicji i inne cenne przedmioty, takie jak płaszcz przeciwdeszczowy, który chroni Cię przed atakami wodnymi. Te ważne bonusy są ukryte w grze i mogą stać się bardzo potrzebne w miarę przechodzenia przez kolejne niebezpieczne poziomy.

Wrogowie wymagają jednego lub dwóch trafień, aby można było wyeliminować nietoperze, czarownice, a nawet szkielety. Białe szczury dadzą Ci niewidzialność, co może być bardzo przydatne podczas biegania przez niebezpieczny teren pełen przeszkód.

Gra również ma miłą dla ucha muzykę i zawiera między innymi piosenkę Cindy Lauper. Inne utwory są również chwytliwe i nadają odpowiednią atmosferę przygody podczas rozgrywki.

Poziom trudności nie jest bardzo duży, ale często trzeba zapamiętać drogę do drzwi, aby uniknąć niepotrzebnego cofania. Niektórzy wrogowie są odporni na ataki, więc strategia odgrywa ważną rolę.



## Metroid

Metroid to gra, która kojarzy mi się z oglądaniem pierwszego filmu z serii Alien. Wędrując po zamkniętych korytarzach pod ziemią, w większości przypadków samotnie masz przeciwko sobie obcych, którzy chcą jak najszybciej pozbawić Cię życia. Ty jako gracz prowadzisz misję

bardziej niebezpiecznych przeciwników.

Na końcu stajemy twarzą w twarz z niegodziwym Mózgiem. Jest to ostatnia bitwa, ale nie ostatnia rzecz, którą musisz zrobić w grze, ponieważ odliczanie rozpoczyna się na nowo, gdy pokonasz zabójczy Mózg.



odnalezienia Mózgu Matki, aby zachować bezpieczeństwo dla galaktyki.

Ścieżka dźwiękowa dodaje również głębi do całego klimatu gry. Nasz bohater jest trochę przestraszony i samotnie podróżuje przez wnętrze obcego świata. Nie ma z kim rozmawiać i nie ma nikogo, kto mógłby podzielić się informacjami na temat wykonanych postępów w rozgrywce. Po pokonaniu dwóch bossów docieramy do nowych obszarów planety. Korytarze zmieniają się z technologicznych na naturalne, wykonane ze skał i kierujące w stronę coraz

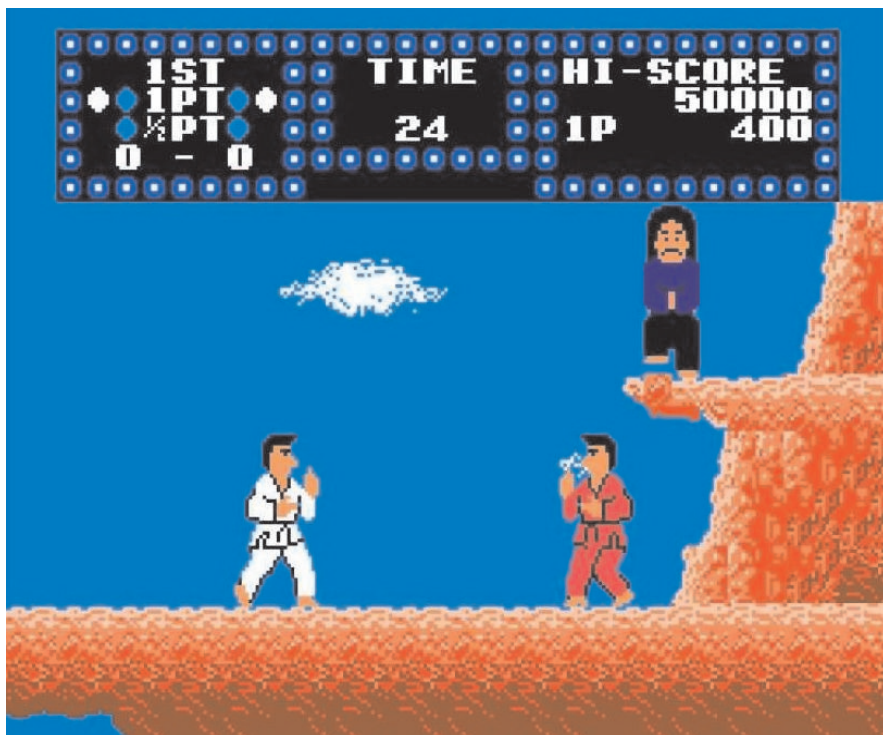
Wiele osób wraca do tego oryginalnego tytułu nie tylko z powodów nostalgicznych. Jest to bardzo dynamiczna gra, do tego dość trudna, ale cały czas zachęca do odnalezienia kolejnych tajemnic, nowych korytarzy i przedmiotów. W trakcie można też zobaczyć kilka drobnych błędów, ale to tylko daje więcej frajdy.

## Over Horizon

8-bitowe konsole były powszechne na rynku, bo były tanie i łatwe w obsłudze, a biblioteka gier zachęcała do zakupu. Over Horizon to strzelanina, która nie posiada pewnych wad R-Type, a dodatkowo ma wiele ulepszeń. Jeśli chodzi o niedoskonałości mam na myśli głównie pozbawienie życia przez jeden strzał. To potrafi denerwować, ale Over Horizon jest grą zupełnie inną, mimo wizualnego podobieństwa.

Rozgrywka jest tutaj bardzo intuicyjna. Masz statek kosmiczny, który





może strzelać. Do dyspozycji są trzy bonusy, a także opcje typu zmiana prędkości po naciśnięciu przysiku Select.

Jednak kilka wyróżniających cech sprawia, że rozgrywka jest tu wyjątkowa, na przykład, możesz strzelać do tyłu. Trzymając oba przyciski ognia jednocześnie, można dostosować położenie dodatkowych pojazdów bezzałogowych w stosunku do głównego statku. Pozwala to na uzyskanie większej ilości sposobów podejścia do walki.

Ciekawą funkcją jest "Edit Mode". Pozwala na dostosowanie wartości odnoszących się do różnych dodatków jakie można zebrać. Jest to sympatyczna funkcja, która może dodać odrobinę powtarzalności do gry. Jest też tryb testowy, który pozwala ćwiczyć umiejętności i zrozumienie mechaniki na poszczególnych poziomach.

Rozgrywka Over Horizon jest taka, jak estetyka całej gry - trudno przeoczyć oczywiste inspiracje, ale wszystko jest bardzo dobrze wykonane. Mamy dżungle kosmiczne, pustynie kosmiczne, a nawet lodowce kosmiczne.

Muzyka nie jest szczególnie wyjątkowa, ale z pewnością jest na przyzwoitym poziomie. Warto zainteresować się Over Horizon, szczególnie jeśli nie mieliśmy okazji grać na sprzęcie Nintendo lub robić to dawno temu.

Na koniec kilka mniejszych gier, ale równie wciągających, choć wykonanych na różnym poziomie.

## Excitebike

Jest to prosta gra wyścigowa, motocrossowa, z widokiem z boku. Mamy wzgórza i rampy, które trzeba przeskakiwać oraz różne inne przeszkody. Można użyć Nitro dla

zwiększenia szybkości, ale trzeba uważać, aby rower nie przegrzewał. Jedną z najlepszych cech tej gry jest to, że pozwala projektować własne tory. Niestety, nie można ich zapisywać. Szkoda też, że nie ma tryb dwuosobowego, ale i tak warto zagrać.

## Flintstones

Jest to gra platformowa oparta na klasycznej animowanej serii. Jako Fred Flintstone, musisz znaleźć kawałki maszyny czasu, aby można było uratować Dina. W grze można znaleźć różne rodzaje broni. Sterowanie działa z lekkim opóźnieniem, a bohater się "ślizga", za to podoba mi się bardzo dobre przewijanie ekranu.

## Karate Champ

Jest to gra walki jeden na jednego, w którą może grać jeden lub dwóch graczy. Za każde trafienie przeciwnika otrzymujesz pełny punkt, albo pół punktu, w zależności od użytego ruchu. Aby wygrać rundę trzeba zdobyć dwa punkty, a cała walka jest rozgrywana w dwóch rundach. Na ekranie widać sędziego, który często obserwuje walkę siedząc na drzewie. W grze jest dziewięć lokalizacji, w których można walczyć. Gra nie ma zakończenia.

Zawodnicy ruszają się nieco dziwnie, a Karate Champ trudno porównywać do odpowiedników na innych platformach jak na przykład International Karate. Urzekła mnie jednak jej prostota, a nawet pewna forma prymitywizmu, bo postacie nie mają dużej ilości klatek, ani w tle nie przegrywa wyjątkowa muzyka, Taki klimat trzeba po prostu lubić.

# Nintendo Classic Mini

KAMIL STOKOWSKI

Premiera konsoli NES firmy Nintendo miała miejsce w 1983 roku. Stała się wielkim hitem, a po latach prawdziwą legendą. Dzisiaj możemy zagrać w kultowe gry także na nowym urządzeniu będącym zminiaturyzowaną wersją starego urządzenia.

Oczywiście w nowej wersji wprowadzono zmiany. Nintendo Classic Mini nie posiada gniazda na cartridge, a gry można uruchamiać z pamięci wewnętrznej. Konsola posiada nowoczesne gniazdo HDMI, a więc bez problemu uzyskamy wysokiej jakości obraz na monitorze lub telewizorze. Jednak największą zmianą jest jednak rozmiar i waga nowej konsoli. Posiada wymiary ok. 12,5 x 10 x 4,5 cm, jest również zaskakująco lekka.

Można byłoby się spodziewać, że urządzenie będzie miało problemy z uruchamianiem niektórych gier lub uzyskaniem odpowiedniej płynności jaką znamy z wyświetlaczy kineskopowych. Trzeba jednak przyznać, że wszystko działa bardzo dobrze, a 30 lat po premierze oryginalnej konsoli, może ona nadać wiele niezapomnianej rozrywki. Standardowo mamy 30 gier do wyboru na czele ze słynną Zeldą i wszystkie działają płynnie. Tytuły są dobrze dobrane i można w nie grać wręcz bez końca.

Konsola wymaga typowego zasilania USB 5V/1A, ale w oryginalnym zestawie jest tylko przewód MicroUSB. W komplecie jest także tylko jeden kontroler, drugi można zakupić oddzielnie. Szkoda, choć są to małe niedostatki nowego NES-a.

Dla mnie pewnym problemem są piksele widoczne na monitorze LCD. Nie wygląda to tak, jak pamiętam z lat '90-tych, ale nie jest to winą firmy Nintendo. Po prostu grafika rysowana na ekranach CRT wygląda inaczej na nowych wyświetlaczach. Z

tego względu wolę jednak oryginalną konsolę podłączoną do starego telewizora, taki zestaw zapewnia prawdziwy klimat.

Wersję Classic Mini polecam jednak wszystkim osobom, które nie są wracać do ekranów CRT, a chcą wygodnie i szybko uruchomić stare hity. W kolejnym numerze magazynu RetroKomp przedstawię bardziej rozbudowany test Classic Mini, jak również porównanie z wersją oryginalną konsoli.

