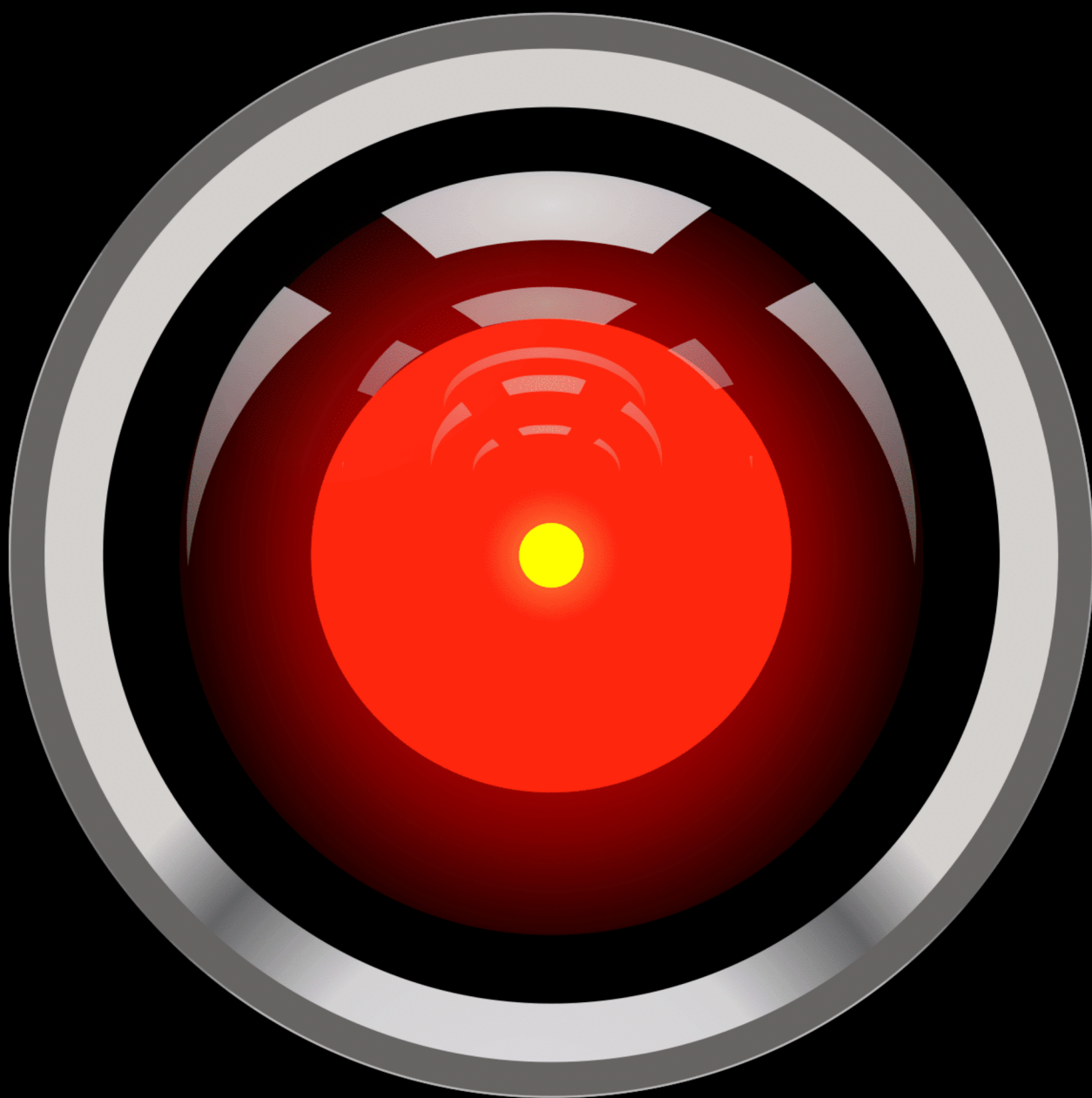


ATARI - COMMODORE - ZX SPECTRUM - AMSTRAD CPC - AMIGA - APPLE

RETROKOMP

Mikrokomputery wczoraj i dziś





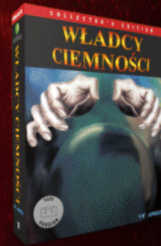
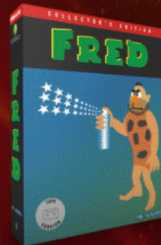
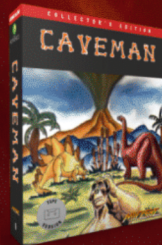
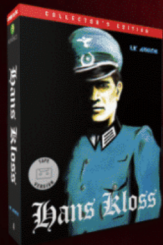
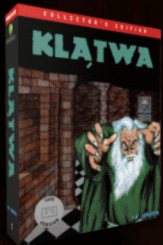
RETRONICS

WHERE BYTES TRAVEL BACK IN TIME...



KOLEKCYJONERSKA EDYCJA KULTOWYCH GIER DLA KOMPUTERÓW 8-BITOWYCH

- ekskluzywna poligrafia
- cyfrowo zrekonstruowane oryginalne wkładki
- 100% wierna kopia oryginału
- dostępne na kasetach i dyskietkach 5,25" w limitowanym nakładzie
- wydanie licencyjne
- do każdego tytułu dołączona wielkoformatowa mapa gry
- instrukcja i karty kodowe w zestawie
- wysoka dbałość o techniczną stronę wydania



• Kolejne tytuły wkrótce!

Tytuły dostępne w przedsprzedaży dla ATARI XL/KE: Kłatwa (LK Avalon), Miecze Valdgira (ASF), Operation Blood (Mirage), Hans Kloss (LK Avalon), Dwie Wieże (ASF), Caveman (Mirage), Fred (LK Avalon), Kult (ASF), Magia (Mirage), Władcy Ciemności (LK Avalon).
Niebawem także tytuły dla Commodore 64/128!



MADE IN POLAND

www.retronics.eu

NASZE WYROBY SĄ DOSTĘPNE W CIĄGŁEJ SPRZEDAŻY NA ALLEGRO.PL



Nowa pasja odkrywania

Zastanawiałem się ostatnio, co trzyma mnie przy spręćie retro. Czy jest to tylko fascynacja z dzieciństwa, która nie chce przeminąć? Założę się, że wielu z Was raz na jakiś czas zadaje sobie pytanie, czy warto dalej zajmować się Commodore, Atari czy Amigą? Myślę, że odpowiedź tkwi tam, gdzie zaczęła się nasza wspólna przygoda - w momencie pierwszego zetknięcia się z komputerem. Czy chcieliśmy wtedy zatrzymać się w rozwoju, czy czekaliśmy z niecierpliwością na nowe osiągnięcia technologiczne?

Warto przypomnieć sobie te chwile, bo szybko okaże się, że fascynacja elektroniką nie miała nic wspólnego z konkretną platformą sprzętową. W czasach, gdy komputer był nieosiągalnym luksusem chcieliśmy mieć jakiegokolwiek sprzęt w domu. Zwykle przypadek sprawiał, że mieliśmy do dyspozycji C64 albo 130XE. Mogła to być także decyzja innego członka rodziny, w każdym razie - cieszyliśmy się z możliwości dostępu do komputera. Zaczynaliśmy odkrywać jego możliwości i najczęściej ten punkt początkowy wyznaczał dalszy rozwój zainteresowań. W moim przypadku na początku był Timex 2048 i zaraz później Commodore Plus/4, który otrzymałem "w spadku" po rodzinie,

która zaopatrzyła się w lepszy 8-bitowy model Atari. Nie da się ukryć, że - niezależnie od chęci wykorzystania potencjału maszyny, jaka stała na naszym biurku - chcieliśmy ją rozbudować, a później także wymienić na kolejny, nowszy model. Parcie do nowoczesności było wszechobecne w naszych głowach, a jednak w pewnym momencie zostało zastąpione schematami, które nakazały nam podążać konkretną ścieżką - od 8-bitowego C64 czy 130XE do Amigi, Atari ST czy Falcona. Dla części osób od razu atrakcyjny stał się IBM PC, inni - jak ja - nie chcieli rezygnować z osiągnięć własnej platformy na korzyść "obcego".

Dzisiaj wracamy do sprzętu retro z różnych powodów, ale bardzo często zatrzymujemy się na nim, bowiem jest to przedmiot, z którym wiążemy pozytywne emocje z przeszłości. Czy nie moglibyśmy robić tego samego z nowymi osiągnięciami techniki? Nie tylko moglibyśmy, ale możemy i często to robimy. Widok sprzętu z różnych epok obok siebie to stały element także w moim domu. Jaki jest więc wspólny mianownik? Chęć poznania czegoś nowego. Każdy z nas może rozumieć to w inny sposób, ale bez zwykłej dziecięcej ciekawości nie da się utrzymać zainteresowania tym, co teoretycznie minęło, a co więcej - uzyskać wartość dodaną. Dlatego nie tracmy tej wspaniałej cechy, która powoduje, iż nasze komputery żyją i będą żyć jeszcze bardzo długo.

W RetroKompie chcemy pokazywać także mniej popularny sprzęt, dlatego

startujemy z nowym działem, poświęconym komputerom Apple. Na początek zabieramy się za modele 8-bitowe i pokazujemy kilka sztuczek w Basicu. Temat będzie kontynuowany, a w niedalekiej przyszłości mam nadzieję na wprowadzenie stałych rubryk przeznaczonych dla użytkowników innych "zapomnianych". Zapraszam do lektury.

Adam Zalepa

RETROKOMP

Mikrokomputery wczoraj i dziś

NUMER 7/2017
(sierpień)

ISSN 2450-5862

Redaktor naczelny: Adam Zalepa

retrokomp@amiga.net.pl

Autorzy tekstów:

Mateusz "Tfardy" Eckert

Tomasz Jaworski

Piotr „Piter” Krużycki

Piotr "Kroll" Mietniowski

Łukasz Nowak

Michał "stRing" Radecki-Mikulicz

Piotr "Sachy" Sachanowicz

Kamil Stokowski

Mariusz Wasilewski

Projekt okładki:

Marzena Bukowska

Korekta: Renata Gralak

Skład: Andrzej Wilczyński

Wydawca:

"A2" Aleksandra Zalepa, Łódź

Wszystkie nazwy oraz znaki handlowe należą do ich właścicieli i zostały użyte wyłącznie w celach informacyjnych.



RETROKOMP 7:

Nowa pasja odkrywania

INTEGRACJA 5

Nowe wydanie książki "Assembler 6502"

Komputery Sharp

F-19 Stealth Fighter - część 1.

Monitory i wektory

Stacje robocze NeXTstation

Analogowy zapis dźwięku

ZX SPECTRUM 20

Wielokolorowa grafika

Dźwięk bez komplikacji

Liczby pseudolosowe

Trudna obsługa klawiatury

AMSTRAD 29

CPC 464 i grafika

Konsola GX4000

COMMODORE 33

Kompresja danych

Interfejs RS-232

Taśmy w praktyce

SuperPET i większa pamięć

Przenoszenie danych na C64

Śledzenie działania

Łączenie programów

ATARI 48

Optymalizacja programów w Atari Basic

DOS XE

Automatyczna numeracja

Kasowanie linii w Basicu

Wykopaliska małe i duże

Jak pisać programy w języku Action?

APPLE 61

Usprawniony katalog

Rysowanie w Basicu

Gry dla Apple II

AMIGA 69

Kontroler Tandem

ScalOS - lepszy blat

Komunikacja pomiędzy procesami

30 lat minęło

Nowe wydanie książki Assembler 6502

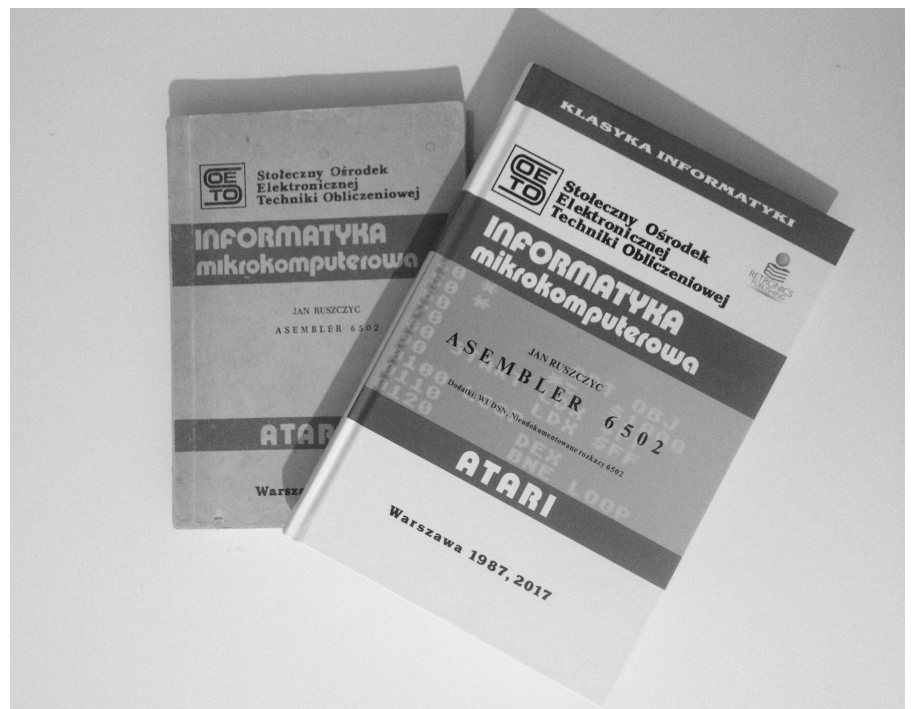
Firma Retronics, producent pokryw antykurzowych do retro komputerów, a także od niedawna wydawca starych gier na małe Atari (recenzję o tej reedycji przeczytacie w RetroKompie 5/2017) wciąż rozwija swoją działalność. Od niedawna można u nich także zamówić wznowienia książek wydawnictwa SOETO traktujące o szeroko pojętej informatyce mikrokomputerowej. Książki te (w tematyce Atari wydano kilkanaście pozycji) były pod koniec lat 80. rozpowszechniane w dość niskich nakładach. Przykładowo „Assembler 6502” Jana Ruszczyca wydano w ilości 1700 egzemplarzy. Do tego jakość ówczesnych wydań pozostawia wiele do życzenia: słaby i kruchy papier, miękka i cienka okładka, marny druk, pojawiające się w tekście błędy. Książki te swoim wyglądem bardziej przypominają uczelniane skrypty niż pełnoprawne pozycje. Co ciekawe sama firma SOETO (obecnie zar-

ządca i administrator nieruchomości w Warszawie) w czasach PRL jako zakład obliczeniowy, zajmowała się m.in. systemami rozliczeń czynszów.

Jerzy Dudek, właściciel firmy Retronics, postanowił, że wyda wznowienia tych książek. Na początek wydano pozycję popularną wśród „Atarowców” (choć skierowaną nie tylko do programistów tego 8-bitowego komputera), czyli wspomniany „Assembler 6502”. Nowe wydanie może jednak nie być obecnie

dostępne (na książki były zapisy, a kilkanaście sztuk trafiło na allegro), ale najpewniej powstanie dodruk. Przygotowywane są też kolejne pozycje: „Poradnik programisty Atari” Wojciecha Zientary, a w planach jest także seria „Mapa pamięci Atari XL/XE” również autorstwa Wojciecha Zientary, „Atari Basic” Wiesława Miguta, czy inne książki opisujące takie komputery jak ZX Spectrum, czy Amstrad CPC.

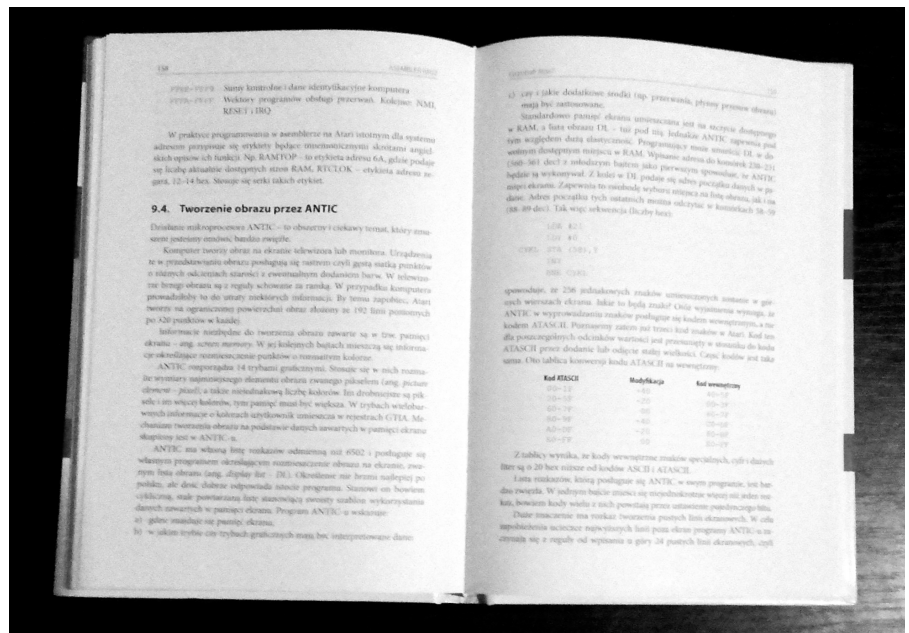
Wszystkie zamówienia, lub deklaracje zakupu można składać na przykład poprzez następującą stronę:



retronics.eu

lub na Facebooku, na profilu wydawcy. Do naszej redakcji trafiła książka Asembler 6502 i tak się składa, że w naszym posiadaniu jest także połówka oryginał z 1987 roku, więc mieliśmy okazję dokładnie się przyjrzeć nowemu wydaniu i porównać obie książki, które dzieli aż 30 lat.

Na pierwszy rzut oka widać, że zwiększył się format. Wcześniej było to 203x147mm, obecnie jest to 242x166mm. W książce z 2017 roku jest też mniej stron (tyczy się to części głównej, bez dodatków). Nowe wydanie ma podobną okładkę z niemal identyczną czcionką jak kiedyś. Dodano tło pod tytułem, ale zachowano kolory: czerwony (w tej barwie są charakterystyczne dla wydań pasy idące przez całą okładkę) i biały. Niedaleko logo dawnego wydawcy umieszczono także logo firmy Retronics. Grzbiet książki doczekał się opisu. Laminowana okładka jest twarda i gruba, z zaokrąglonym grzbietem. Dwie zupełnie odrębne epoki jeśli chodzi o wygląd i jakość. Dodatkowo książka jest szyta nitią (ówczesne wydania miały metalowe zszywacze spinające całość). W środku nowego wydania znajdziemy dobrej jakości papier kredowy i wyczuwalną pod opuszkami palców czcionkę. Wszystkie tabele, rysunki,



schematy czy tablice zostały stworzone na nowo, przez co stały się bardziej czytelne. Tytuły i punkty są pogrubione, przez co książkę czyta się o wiele łatwiej. Na górze każdej ze stron jest opis rozdziału w którym się obecnie znajdujemy, przez co lepiej można się w książce odnaleźć. Spis treści przeniesiono na początek, co też w naszej ocenie było dobrym pomysłem. Oczywiście ulepszony został także skład tekstu. Treść książki również została poprawiona (w oryginale występowało mnóstwo błędów), chociaż należy wspomnieć, że erratę oryginału, można już było odnaleźć w internecie zapisaną w pliku PDF. Warto dodać, że w nowym wydaniu zostały dodane dwa zupełnie nowe dodatki wydrukowane w kolorze. Są to: „Zintegrowane środowisko programistyczne WUDSN” Marcina Sochackiego, oraz „Rozkazy nieudokumentowane 6502C – dodatek dla zaawansowanych” Krzysztofa Dudka oraz Piotra Fusika. Jest to ciekawe urozmaicenie i z pewnością wzbogaci zainteresowanych o nową wiedzę. Co do samej treści, to nie będziemy się o niej rozpisy-

wać. Po pierwsze nikt z nas póki co nie przebrnął przez tę ogromną kopalię wiedzy o assemblerze, po drugie książka nie jest nowa i z pewnością wiele osób ją zna. Nie specjalnie też czujemy się jako recenzenci w tak specjalistycznej dziedzinie. Trzeba by doświadczonego programisty, który mógłby to rzetelnie ocenić.

Czy potrzeba nam takich książek? Patrząc obiektywnie, mimo iż nie ma na nie zbyt dużego rynku zbyt uważamy, że warto uzupełniać o nie swoje biblioteczki. Stare wydania można co prawda czasem zakupić na portalu aukcyjnym, ale ich stan często bywa fatalny. Nowe wydanie pozbawione jest zaś błędów i zawiera ciekawe dodatki. Uważamy również, że żaden pdf odczytany na tablecie nie zastąpi drukowanej książki, chociaż to akurat bardziej subiektywne odczucie. Kwota 89 PLN za książkę nie jest niska, ale profesjonalne i eleganckie wydanie w pełni to rekompensuje. Polecamy ją doświadczonym programistom, a także osobom zaczynającym dopiero naukę programowania na retrokomputerach. No i oczywiście czekamy na kolejne pozycje od Retronics, a w szczególności na znany chyba każdemu użytkownikowi komputera „Atari Basic”!

**Michał "stRing"
Radecki-Mikulicz**



F-19

Stealth Fighter

część 1.

Gry symulacyjne są dzisiaj raczej w odwrocie, a jest to jeden z moich ulubionych gatunków. W przeszłości zagrywałem się w tytuły takie jak choćby Gunship 2000 czy Coala. Trzeba przyznać, że większość z nich traktowałem raczej zręcznościowo, a elementy realizmu nie miały większego znaczenia. Z czasem zmieniłem zdanie, dlatego chciałbym przypomnieć grę F-19 Stealth Fighter, która pod względem stopnia skomplikowania w wielu miejscach nie ustępuje nowszym produkcjom. Zagłębimy się w szczegóły rozgrywki rozpoczynając od elementów kokpitu naszego samolotu. Gra ukazała się w wersjach dla Commodore 64, Atari ST, Amigi, a nawet ZX Spectrum, co w pewnym momencie mojego "życia retro" było nie lada niespodzianką.

Gra oferuje szeroki zestaw misji rozgrywających się w różnych częściach świata - Libia, Zatoka Perska, Europa Środkowa i Skandynawia. W zależności od stopnia zaawansowania

gracz może w różny sposób wpływać na czynniki decydujące o realizmie lotu. Możemy przykładowo ułatwić sobie start lub lądowanie. Nasz samolot w powietrzu zachowuje się całkiem sprawnie, chociaż oczywiście trzeba powiedzieć o zastosowaniu dużych uproszczeń. Na uwagę zasługuje ogromna ilość broni, różnorodność jednostek przeciwnika oraz obecność maszyn cywilnych.

Większość opcji możemy wybierać joystickiem lub myszką, albo za pomocą klawiszy kursora. Większość funkcji wpływa na stopień trudności rozgrywki, między innymi obszar misji oraz rodzaj konfliktu. Zasady prowadzenia walki określają jakie

cele na terytorium przeciwnika mogą być atakowane. W epoce rozpowszechnionej broni nuklearnej wojna nie powinna być nieograniczona, dlatego używane środki są limitowane. Stosowane metody są dość skomplikowane i warto zapoznać się ze wszystkimi komunikatami obecnymi w grze.

Dla każdego regionu w grze możemy wybrać ogólny rodzaj zadania, na przykład niszczenie lub rozpoznawanie celów naziemnych. Do dyspozycji mamy opcje treningowe, dzięki czemu możliwe jest nauczanie się obsługi bez presji czasu. Wybrany przez nas poziom wpływa zarówno na wielkość uszkodzeń powodowa-





nych przez pociski nieprzyjaciela, jak i na skalę trudności przy lądowaniu. Ustalany jest także podstawowy zestaw uzbrojenia, jednak można go swobodnie zmieniać.

Ekran gry zależy od komputera, wygląda nieco inaczej na C64 niż na Amidze, ale podstawowe elementy i zasady ich działania są takie same. Po lewej stronie widać wskaźnik prędkości wyświetlający wartości w węzłach, czyli milach na godzinę. Czarna część sygnalizuje prędkość, przy której samolot traci sterowność. Będzie ona zmieniać się podczas różnych manewrów, a więc należy go obserwować. Listwa po prawej stronie pokazuje wysokość, tym razem jest podawana w stopach lub tysiącach stóp. Na górnej skali widoczny jest bieżący kurs w stopniach. Wyróżniony punkt wskazuje kurs, jaki gracz powinien obrać.

Celownik w centralnej części ekranu wyszukuje ciągle potencjalne cele znajdujące się w naszym zasięgu. Jeśli używamy pocisków zdalnie sterowanych, prostokątny celownik zmieni się w okrągły w sytuacji, gdy cel zostanie odnaleziony i uchwycony poprzez system naprowadzania. Gdy zbliżamy się do celu, celownik zmienia kolor na czerwony, co oznacza,

jenia, uszkodzenia, dane o punktach nawigacyjnych oraz identyfikację celów. Po prawej stronie znajdują się oznaczenia uzbrojenia, dzięki którym wiemy jaki jest aktualnie aktywny rodzaj broni. Na monitorze kolorowym wyświetlane są natomiast mapy systemu nawigacyjnego. Północ zawsze widoczna jest "na górze". Mamy również wielokolorowe punkty, z których żółte oznaczają punkty nawigacyjne, a biały to nasz samolot. Aktywny punkt oznaczony jest przez migotanie.

Wskaźnik EMV informuje o bieżącej widzialności samolotu określanej radiolokacyjnie. Większa ilość "zapalonych" elementów oznacza wyższą widzialność. Poniżej znajdziemy wskaźniki systemów obronnych. W



że możemy liczyć na dużo większą celność. Strzelać można w momencie, gdy odległość od celu jest mniejsza niż 3 mile.

Tablica przyrządów to w większości wielofunkcyjne monitory oraz wskaźniki listwowe, które uzupełnione są o wyświetlacz na przedniej szybie. Pokazuje on automatyczne komunikaty nadchodzące przez radio. Jego stan można zmieniać tak, aby wskazywał stan uzbro-

momencie uaktywnienia odpowiedni wskaźnik zapala się, a gdy zostanie przygaszony - urządzenie przerwało pracę lub zostało wyłączone ze względu na konieczność chłodzenia.

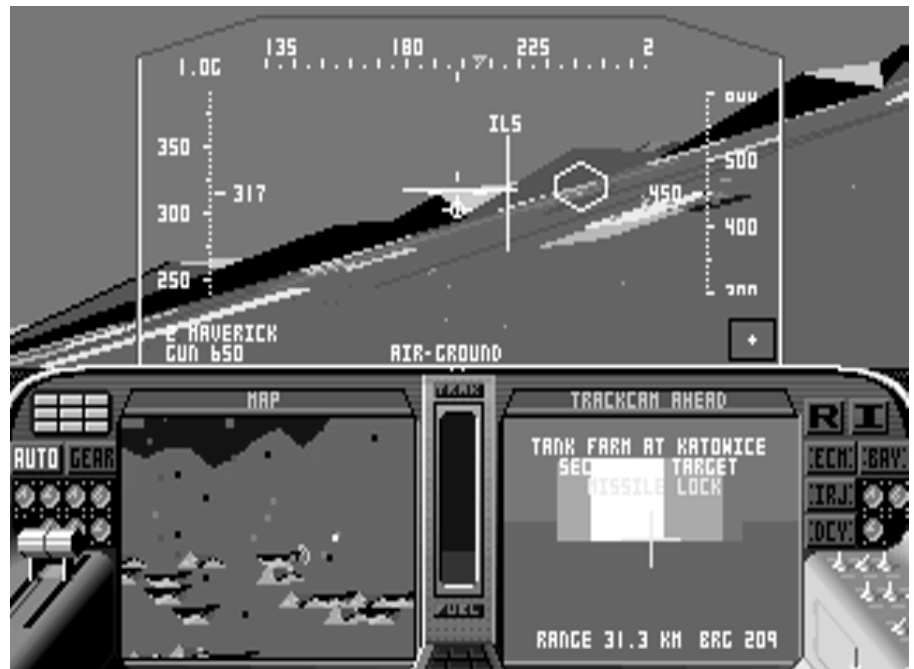
Bardzo ważnym elementem jest wskaźnik zagrożenia, który informuje o położeniu radarów i źródeł "obcego" promieniowania podczerwonego. Każdy rodzaj obiektu zaznaczany jest innym kolorem, natomiast punkty

oznaczające samoloty przeciwnika zmieniają barwę w zależności od aktualnej wysokości. Jeśli punkt jest niebieski - samolot jest wyżej od naszego, gdy jest zielony - samolot jest niżej. Kolor biały oznacza pocisk wystrzelony przez obcy samolot.

Na ekranie mamy również wskaźniki ostrzegawcze RWR i IWR, które mówią o zagrożeniach spowodowanych przez urządzenia naprowadzające za pomocą radaru lub podczerwieni. Barwa niebieska oznacza, że nasz samolot nie jest widoczny, żółta - wręcz przeciwnie. Gdy miga kolor czerwony możemy być pewni, że w naszym kierunku wystrzelono pocisk.

Wskaźniki zużycia paliwa widoczne są w centralnej i dolnej części ekranu. Kiedy zbiorniki są pełne, wskaźnik ma położenie "na górze". Bieżące zużycie paliwa zależy od aktualnej siły ciągu. Paliwo jest zużywane wolniej także wtedy, gdy nasz samolot uzyska większą wysokość. Możemy napotkać takie niespodzianki jak przeciek zbiornika lub konieczność przetankowania paliwa przenoszonego w komorach uzbrojenia do głównego zbiornika. To ostatnie wykonujemy w identyczny sposób jak przy używaniu pocisków.

Po lewej stronie umieszczony jest wskaźnik prędkości wznoszenia lub opadania, czyli tak zwanej prędkości pionowej. Jeśli znajduje się on powyżej środkowej linii oznacza to, że samolot się wznosi, gdy będzie poniżej - samolot opada. Lewa strona



ekranu to także cztery wskaźniki oznaczające otwarte drzwi w komorze uzbrojenia (WB), wysunięte podwozie (LG), wysunięte klapy hamulcowe (SB) oraz wysunięte klapy strumieniowe (FLP). Klapy hamulcowe powodują zmniejszenie prędkości, natomiast strumieniowe zwiększają siłę nośną, ale jednocześnie zmniejszamy prędkość.

Podczas rozgrywki samolotem kierujemy oczywiście za pomocą joysticka. Przyciągnięcie go do siebie spowoduje wznoszenie, odsunięcie od siebie - opadanie. Klawisz Fire służy zarówno do odpalania pocisków, jak i strzelania z działka oraz zrzucania bomb. Wszystko zależy od aktualnie włączonego uzbrojenia.

Wiemy już z grubsza jak wygląda ekran gry. Trzeba przyznać, że jest dość skomplikowany, a nie omówiliśmy wszystkich szczegółów. Zauważmy jak rozbudowana może być symulacja działająca nawet na komputerach 8-bitowych. Gdy nauczymy się obsługi "F-19 Stealth Fighter" okaże się piekielnie wciągającym tytułem, który można uruchomić na wielu platformach retro.

Omówione elementy wystarczą do wykonania pierwszej misji, czyli startu samolotu. Podstawy pilotażu to jednak temat na kolejny dłuższy artykuł, dlatego przedstawię go w następnym odcinku.

Kamil Stokowski

ATARI
SERVIS

Monitory i wektory

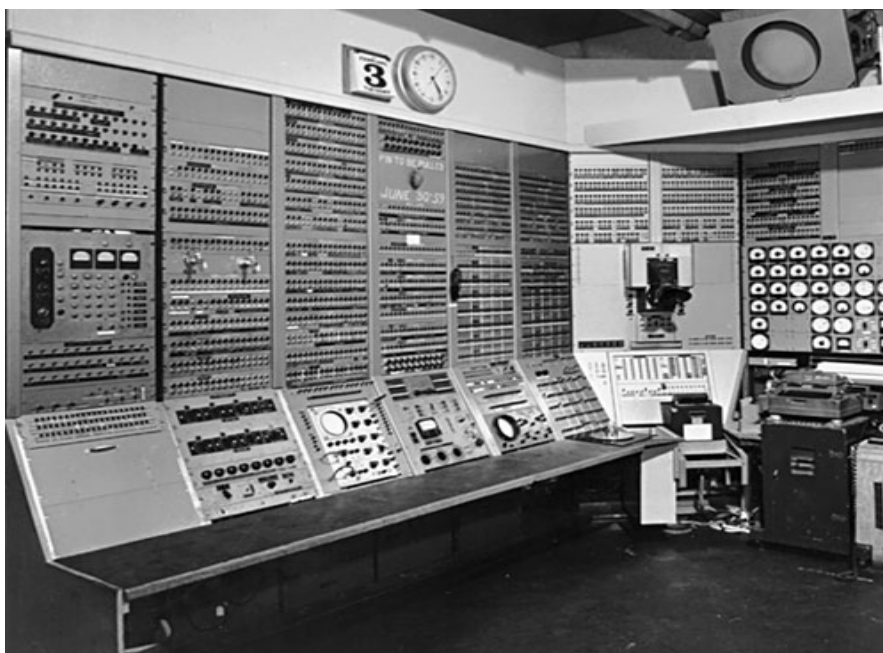
Od początku istnienia informatyki wykonywano niedoskonałe rysunki na urządzeniach drukujących. Pierwsze komputery korzystały z synchroskopu, który wyświetlać słowa na ekranie oscyloskopu. W maszynach z lat '70-tych używane były nawet dalekopisy lub elektryczne maszyny do pisania. Dopiero dekadę później doczekaliśmy się interfejsu wyświetlanego na ekranie monitora.

Komputer Whirlwind opracowany w 1950 roku w MIT miał wyjściowe urządzenie wyświetlające z lampą CRT sterowaną przez komputer, przeznaczoną dla operatora oraz dla tworzenia kopii za pomocą aparatu fotograficznego. Na początku lat '60-tych Ivan Sutherland opracował system rysujący o nazwie Sketchpad. Wprowadził on struktury danych dla zapamiętywania hierarchii symboli oraz opracował metody interakcji za pomocą klawiatury i pióra świetlnego do dokonywania wyborów, wskazywania i rysowania. Sutherland sformułował wiele podstawowych idei i metod, które wciąż są stosowane. Można uznać, że były to początki no-

woczesnej grafiki interaktywnej. W tym samym czasie producenci komputerów, samochodów i sprzętu wojskowego zaczęli doceniać ogromny potencjał systemów CAD i CAM w zakresie automatyzacji rysowania i innych czynności wymagających dużej ilości grafiki. Systemy CAD z General Motors do projektowania samochodów oraz Itek Digitek do projektowania soczewek były pionierskimi programami, które pokazały użyteczność interakcji graficznej w inżynierii.

W tamtych czasach informacje wprowadzano za pomocą kart perforowanych i tak samo odczytywane były rezultaty. Wiązano jednak nadzieje w związku z wprowadzeniem interaktywnej komunikacji na linii użytkownik - komputer. Grafika miała stać się integralną częścią ogromnie przyspieszonego cyklu projektowania. Grafika interakcyjna, ze względu na wysokie koszty, pozostawała wówczas poza zasięgiem większości potencjalnych użytkowników, z wyjątkiem organizacji najbardziej zaawansowanych technologicznie.

Do początku lat '80-tych grafika komputerowa była wąską specjalizacją, głównie ze względu na koszty sprzętu i niewielką liczbę programów użytkowych korzystających z tej formy komunikacji. Później komputery osobiste z wbudowanymi

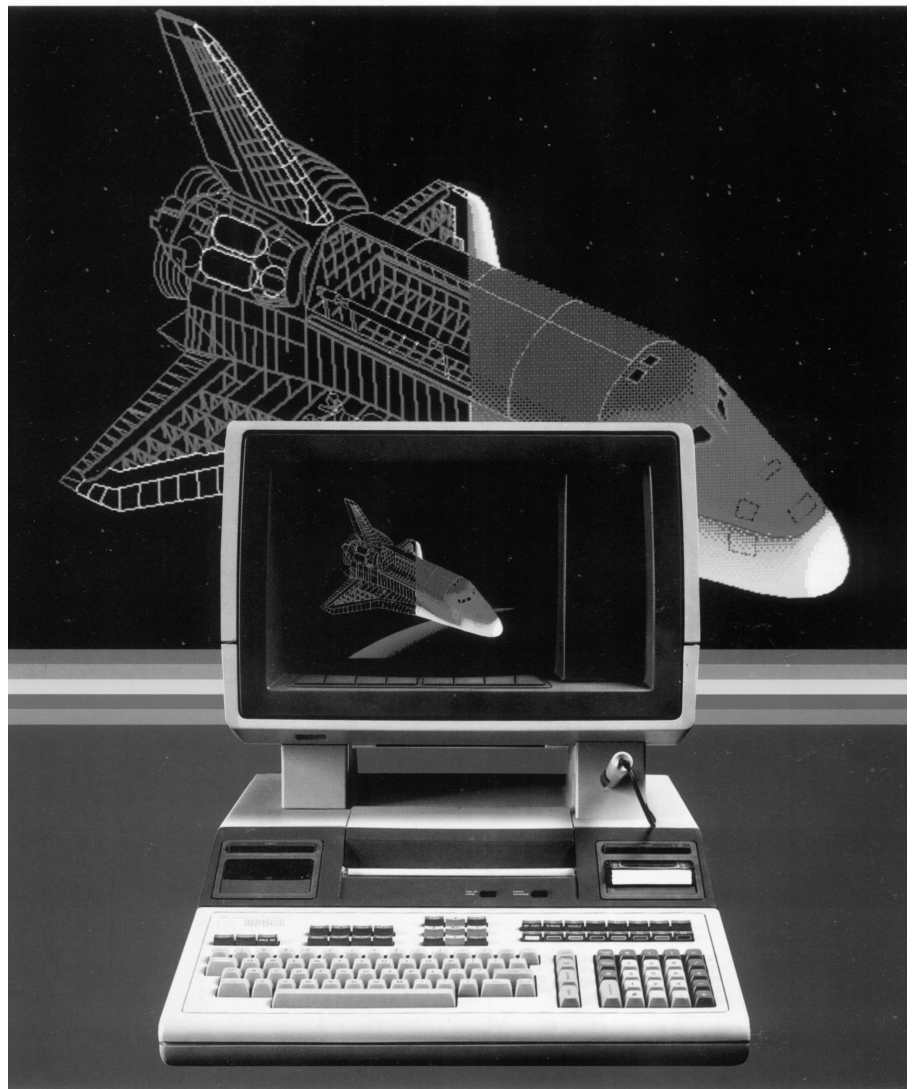


rastrowymi urządzeniami spopularyzowały korzystanie z grafiki z mapą bitową. Jest to zero-jedynkowa reprezentacja prostokątnej tablicy punktów na ekranie, nazywanych pikselami. Dziś wszyscy to znamy z codziennego kontaktu z komputerem.

Wkrótce po tym, jak grafika bitmapowa stała się dostępna, nastąpiła eksplozja łatwych w użytkowaniu programów. Nowy interfejs użytkownika wykorzystujący grafikę umożliwił milionom nowych użytkowników korzystanie z produktów takich jak arkusze kalkulacyjne, edytory tekstów i programy graficzne. Bezpośrednie manipulowanie obiektami na zasadzie ich wskazywania zastąpiło w znacznym stopniu wypisywanie tajemniczych poleceń stosowanych we wcześniejszych komputerach, na przykład HP 9845C, który mimo ograniczeń był już określany mianem "graficzny". Obecnie prawie wszystkie programy użytkowe powszechnie wykorzystują grafikę w interfejsie użytkownika oraz dla wizualizacji i manipulowania obiektami.

Urządzenia wyświetlające opracowane w połowie lat '60-tych i wykorzystywane do połowy lat '80-tych są określane jako monitory wektorowe, kreskowe, rysujące odcinki albo kaligraficzne. Określenie "wektor" jest używane jako synonim odcinka, natomiast kreska jest krótkim odcinkiem i znaki są tworzone z ciągów takich kresiek. Typowy system wektorowy zawiera procesor monitora przyłączony jako zewnętrzne urządzenie wejścia/wyjścia do centralnej jednostki przetwarzającej, pamięci buforowej monitora i CRT.

Istotą systemu wektorowego jest to, że strumień elektronów jest odchylany od jednego końca odcinka do drugiego, zależnie od kolejności poleceń wyświetlania. Jest to tak zwana metoda przeszukiwania przypadkowego. Światło emitowane przez luminofor zanika w ciągu dziesiątek lub setek mikrosekund, dlatego procesor monitora musi cyk-



licznie powtarzać listę wyświetlanych elementów w celu odświeżenia luminoforu przynajmniej 30 razy na sekund, co daje nam 30 Hz, po to, aby uniknąć migotania.

Opracowanie na początku lat '70-tych taniej grafiki rastrowej wykorzystującej technologię telewizyjną miało znacznie większy wpływ na rozwój grafiki komputerowej niż jakakolwiek inna technologia. Monitory rastrowe zapisują wyświetlane elementy (np. odcinki, okręgi, znaki, wypełnione w sposób ciągły lub wzorami obszary) w pamięci ekranu w postaci pikseli tworzących określony kształt. W niektórych monitorach sprzętowy kontroler monitora odbiera i interpretuje sekwencje poleceń wyjściowych, w

prostszych systemach kontroler monitora istnieje jako element programowy biblioteki graficznej, a pamięć ekranu jest częścią pamięci procesora, która może być odczytywana przez podsystem wyświetlania obrazu (a więc sterownik), który tworzy obraz na ekranie.

Kompletny obraz na monitorze rastrowym jest tworzony na bazie rastra, czyli zbioru poziomych linii składających się z pikseli. Raster jest zapamiętywany jako tablica pikseli reprezentujących całą powierzchnię ekranu. Obraz jest tworzony sekwencyjnie przez sterownik wyświetlania, linia po linii z góry na dół i potem - ponownie od góry. Dla każdego piksela natężenie strumienia jest tak

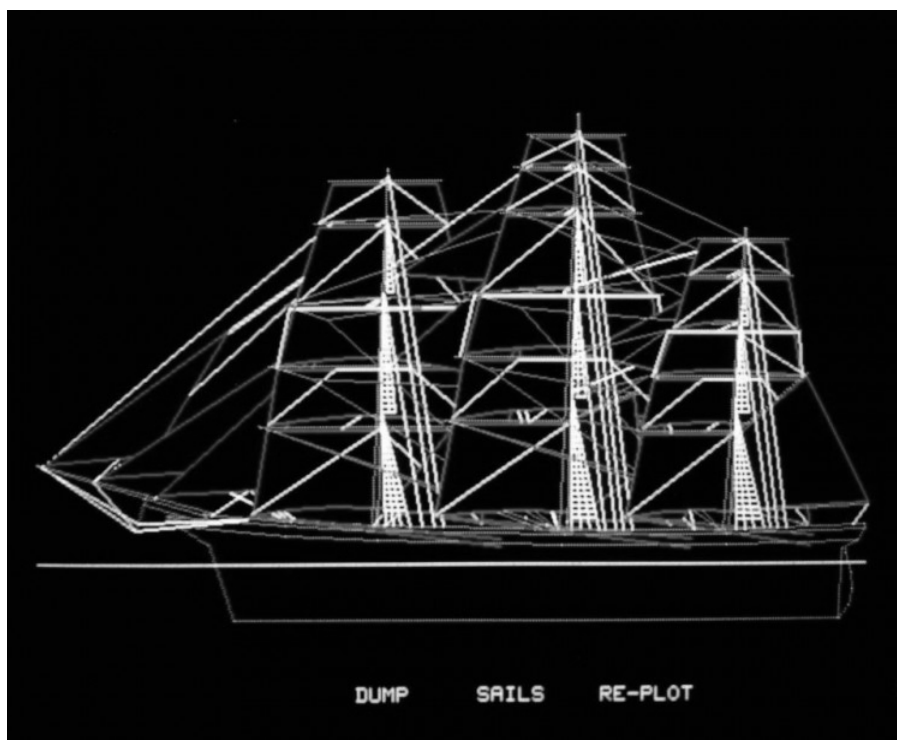
ustawiane, aby odzwierciedlić jasność piksela. W systemach barwnych są sterowane trzy strumienie – po jednym dla każdej barwy podstawowej: czerwonej, zielonej i niebieskiej – zgodnie ze specyfikacją trzech składowych barwy dla każdej wartości piksela. W systemie rastrowym cała siatka (np. 1024 linie po 1024 piksele) musi być bezpośrednio zapamiętana.

Na początku lat '70-tych brak tanich półprzewodników potrzebnych do budowy pamięci dla mapy bitowej stanowił istotne ograniczenie rozwoju grafiki rastrowej i uniemożliwiał jej zdobycie dominującej pozycji. Dwupoziomowe monitory CRT (monochromatyczne) rysowały obrazy biało-czarne, czarno-zielone albo czarno-pomarańczowe.

W dwupoziomowej mapie bitowej każdemu pikselowi jest przyporządkowany 1 bit i mapa bitowa ekranu o rozdzielczości 1024 na 1024 piksele liczy 2^{20} bitów czyli około 128 KB. W prostych systemach barwnych jest 8 bitów na piksel, dzięki czemu dostępnych jest równocześnie 256 barw. W bardziej rozbudowanych systemach są 24 bity na piksel i istnieje możliwość wyświetlenia ponad 16 milionów barw.

Dostępne są również pamięci obrazu z 32 bitami na piksel, w których 24 bity są przeznaczone na reprezentowanie barwy, a 8 jest wykorzystywanych do celów sterowania (np. jasności). W zasadzie właściwie pojęcie mapy bitowej powinno odnosić się tylko do systemów dwupoziomowych, gdzie jest 1 bit na piksel. Dla systemów, gdzie każdemu pikselowi przyporządkowanych jest wiele bitów, używamy pojęcia mapy pikselowej.

Główne zalety grafiki rastrowej w porównaniu z grafiką wektorową to niższy koszt i możliwość wyświetlania obszarów wypełnionych jednolitą barwą lub wzorami – co jest ważne przy tworzeniu realistycznych obrazów obiektów 3D. Główna wada systemów rastrowych w porównaniu



z systemami wektorowymi jest natomiast związana z dyskretną naturą reprezentacji piksela. Obiekty, takie jak odcinki i wielokąty, są określane przez parametry ich wierzchołków i muszą być odwzorowane w pamięci obrazu za pomocą pikseli.

Ten proces odwzorowania, czy też konwersji, jest określany jako rasteryzacja. W komputerach, gdzie mikroprocesor jest odpowiedzialny za całą grafikę, rasteryzacja jest wykonywana proprogramowo. Inna wada systemów rastrowych wynika z natury rastra. Podczas gdy system wektorowy może rysować ciągłe gładkie odcinki i gładkie krzywe w zasadzie od dowolnego punktu na ekranie do dowolnego innego punktu, system rastrowy może wyświetlać matematycznie gładkie linie, wielokąty i brzegi elementów takich jak okręgi i elipsy tylko na zasadzie ich aproksymacji (przybliżenia) za pomocą pikseli należących do siatki rastra.

Ten mechanizm może powodować powstawanie problemu "schodków" albo "ząbków", w zależności od użytej rozdzielczości ekranu. Wizual-

ne zakłócenie jest konsekwencją błędów próbkowania określanego w teorii przetwarzania sygnałów jako aliasing. Takie zakłócenia pojawiają się wówczas, gdy funkcja ciągłej zmiennej zawierająca ostre zmiany jasności jest aproksymowana za pomocą wspomnianych dyskretnych próbek.

We współczesnej grafice komputerowej są stosowane metody likwidowania zakłóceń, co nazywamy słynnym antyaliasingiem. Metody określają gradację jasności sąsiednich pikseli na granicach kształtów. Z upływem lat ulepszono również metody wprowadzania informacji. Niewygodne i delikatne pióro świetlne, niegdyś synonim nowoczesności, zastąpiono wszechobecną myszką, a następnie pojawiły się ekrany czułe na dotyk. Dostępne są urządzenia wejściowe, które określają nie tylko położenie współrzędnych na ekranie, ale również współrzędne 3D, a nawet więcej wymiarowe wartości wejściowe. To naprawdę bardzo ciekawa historia, o której napiszemy więcej już niebawem.

Mariusz Wasilewski

Komputery Sharp

W 1978 roku firma Sharp zaczęła sprzedawać swój pierwszy domowy mikrokomputer: MZ-40 K, oparty na 4 bitowym procesorze. Był to zestaw do samodzielnego montażu i z trudnością przypominał komputer. Było to raczej urządzenie mikroprocesorowe do programowania i odtwarzania muzyki oraz prostych, wbudowanych gier losowych. Sharp nie dostarczał żadnej dokumentacji dotyczącej programowania w kodzie maszynowym. Do kogo miało to urządzenie trafić? Nie wiadomo. Być może Sharp chciał przetestować rynek. MZ-40 K sprzedawany w Japonii i Niemczech cieszył się mimo wszystko sporym zainteresowaniem. Dyskontując jego sukces, a także popularność takich maszyn jak Commodore PET, postanowiono zbudować komputer z prawdziwego zdarzenia.

MZ-80K

W ten sposób powstał MZ-80K. Komputer posiadał wbudowany czarno-biały monitor i magnetofon kasetowy. W pamięci ROM nie było Basica, ale trzeba było go wgrać z kasety. Zaczynał on liczną serię komputerów opartą na procesorze Z80. Prawie 16-kilogramowy MZ-80B to czwarte wcielenie modelu MZ-40K.

Po raz pierwszy w serii wprowadzono możliwość uzyskania grafiki w "wysokiej" rozdzielczości 320 x 200 punktów. Komputer ten pokazano w Europie po raz pierwszy w kwietniu 1981 roku na targach komputerowych w Hannoverze. Jednak jego wysoka cena spowodowała, że nie był w stanie rywalizować z Commodore czy Apple. W Japonii rodzina MZ była

przez długie lata najpopularniejszymi komputerami domowymi.

MZ-2500

Kolejny ciekawy model to MZ-2500, nazywany również ze względu na swoje możliwości - Super MZ. Był jednym z ostatnich komputerów Sharp wyposażonych w ośmiobitowy procesor Z80. Jest on w prostej linii następcą posiadanego przez mnie Sharp'a MZ-80B.

Sprzedawany był w kilku wersjach:

- 2511 z magnetofonem i jedną stacją 3,5" dyskietek
- 2520 z dwoma stacjami 3,5" dyskietek
- 2521 z magnetofonem i dwoma stacjami dyskietek

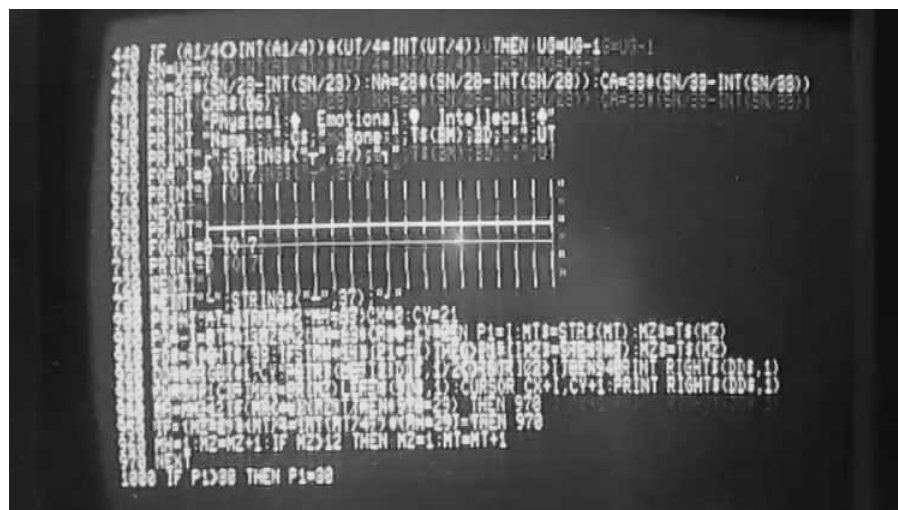
Specyfikacja techniczna komputera Sharp MZ-80B:

Procesor:	LH0080A (klon Z80A) z zegarem 4 MHz
Pamięć:	RAM 32 KB (maks. 64 KB)
Pamięć VRAM:	2 KB
Pamięć ROM:	2 KB (generator znaków), 2 KB (Program Loader)
Grafika:	320 x 200 pikseli
Tekst:	tryb 80 x 25 lub 40 x 25
Kolory:	tryb monochromatyczny, wbudowany monitor 9-calowy
Dźwięk:	1 kanał, wbudowany głośnik
Porty:	Centronics, RS-232C, IEEE 488, interfejs stacji dyskietek (opcjonalne)
System:	CP/M wczytywany z dysku Basic wczytywany z taśmy



Oparty jest on na jednym lub więcej procesorach Z80 pracujących z częstotliwością 6MHz. Ma podstawowy system operacyjny oznaczony "6Z002 V1.0C". Podstawowy system operacyjny 6Z002 jest w stanie obsługiwać pamięć w blokach po 8 KB. Teoretycznie może obsłużyć 64 bloki, co daje całkowitą pojemność 512KB pamięci RAM, ale w praktyce wykorzystywane są tylko 52 bloki, co daje łączną pojemność 416 KB.

Pierwsze 48 bloków obejmują standardowy obszar 128 KB pamięci RAM, która jest rozszerzalna do 256K oraz 64 KB pamięci VRAM z możliwością rozbudowy do 128 KB. Pozostałe 4 bloki zawierają wbudowany ROM o pojemności 32 KB. Całkowita pamięć to 384 KB RAM plus 32 KB ROM, czyli razem 416 KB.



Na płycie głównej MZ-2500 jest jeszcze jeden wolny slot oznaczony "MZ-1M10", który miał pasować do karty "16-Bit Kit" wykorzystującej bloki pamięci od 3B do 3F. To wyjaśnia, dlaczego oprogramowanie BASIC jest teoretycznie zdolne do przełączania do 64 bloków i 512K pamięci, w praktycznie jego działanie ogranicza się do wspomnianych wcześniej 52 bloków. Komputer był dostarczany z monochromatycznym monitorem o dość dziwnym oznaczeniu "MD-9P1".

Komputer ma bardzo ciekawą obudowę, bardziej przypominającą na pierwszy rzut oka odtwarzacz kasetowy niż komputer. Zestaw to solidna metalowa obudowa oraz oddzielna klawiatura przypominająca układ PC XT. Sharp MZ-2500 to następny model sprzedawany w Japonii w 1985 roku, który nigdy nie dostępny w szerszej sprzedaży.

Był również określany jako maszyna "SUPER-MZ". Był to ostatni komputer z serii MZ i trzeba przyznać, że jednym z najbardziej wyrafinowanych sprzętów z procesorem Z80.

Specyfikacja techniczna komputera Sharp MZ-2500:

Procesor:	Z80B z zegarem 6 MHz
Pamięć RAM:	128 KB (maks. 256 KB)
Pamięć VRAM:	64 KB (maks. 128 KB)
Pamięć ROM:	32 KB (maks. 256 KB)
Grafika:	640 x 400, 640 x 200, 320 x 200
Tekst:	80 x 25 (2 kolory), 40 x 25 (4 kolory), 20 x 25 (16 kolorów)
Kolory:	256 (ilość jednocześnie wyświetlanych kolorów zależy od wielkości pamięci VRAM)
Dźwięk:	3 kanały
Porty:	Extension, Klawiatura, RS-232
System:	CP/M

MZ-700

Chyba jeszcze mniej znanym sprzętem jest seria MZ-700. Posiadał procesor Z80A oraz pamięć 64 KB. Model MZ-711 mógł być podłączany bezpośrednio do zwykłego telewizora, opcjonalnie dostępne były również 12-calowe monochromatyczne monitory MZ-1D04 oraz 14-calowe monitory kolorowe MZ-1D05. Cała seria była reklamowana jako komputery do dalszej rozbudowy.

Przykładem innowacji Sharp'a może być MZ-731 posiadający jedyną w swoim rodzaju obudowę z wbudowanym magnetofonem i ploterem, która zajmowała niewielką ilość miejsca na biurku. Nawet dzisiaj zastosowane rozwiązanie wydaje się nowoczesne. Klawiatura zawierała 5 klawiszy funkcyjnych, które pozwalają na wygodne zapamiętanie 10 ostatnio potrzebnych poleceń. Klawiatura zawiera także klawisze wywołujące tryby edycji tekstu: wstawiania lub usuwania oraz powrotu do pozycji początkowej.



Specyfikacja techniczna komputera Sharp MZ-700:

Procesor: Sharp LH-0080
(zgodny z Zilog Z80 A)

Zegar: 4 MHz

Pamięć RAM: 64 KB

Pamięć VRAM: 2 KB

Pamięć ROM: 2 KB

Tryb tekstowy: 40 x 25

Ilość kolorów: 8

Dźwięk: 1 kanał, 3 oktawy

Klawiatura: 69 klawiszy, 5 klawiszy funkcyjnych, 4 klawisze kursora

Porty: parallel, joystick (2), Z80 Bus, magnetofon, RGB

System: CP/M wczytywany ze stacji dyskietek (opcjonalnie)

Stacje robocze NeXTstation

NeXTstation to uniksowa stacja robocza, produkt firmy NeXT założonej w 1985 roku przez Steve'a Jobsa. Choć nie odniosła sukcesu rynkowego to w znacznym stopniu przyczyniła się do wytyczenia nowych kierunków w konstrukcji komputerów i tworzeniu systemów operacyjnych. Dlatego warto przypomnieć ten mało popularny sprzęt.

W 1985 roku Steve Jobs pozbawiony wpływu na zarządzanie Apple odszedł i kosztem 7 milionów dolarów założył nową firmę NeXT. Pozew sądowy Apple i ugoda zawarta w styczniu 1986 roku pomiędzy firmami ograniczyły możliwości działania NeXT do rynku stacji roboczych. W 1988 roku NeXT zadebiutował ze swoim pierwszym produktem. Komputer o nazwie NeXTcube, bo o nim mowa, był wyjątkowy z wielu względów.

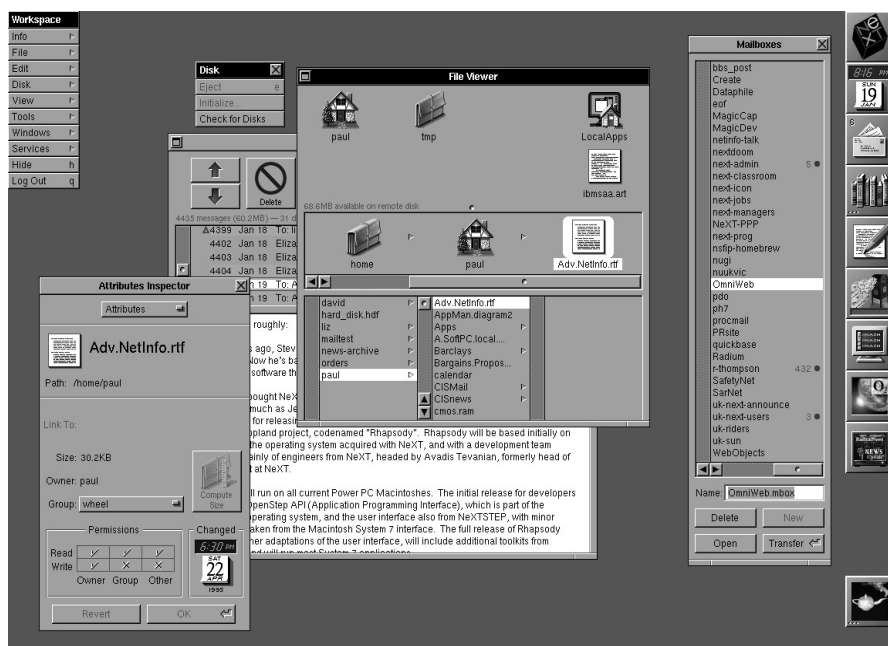
Większość komputerów w tym okresie była zaopatrzona w zestaw stacji dyskiety + dysk twardy od 20 do 40 MB. Większe napędy były bardzo drogie. W nowym komputerze postanowiono zastąpić to jednym, bardzo pojemnym medium. Zastosowano debiutujący na rynku napęd magneto-optyczny firmy Can-

on o zawrotnej wtedy pojemności 256 MB. Sercem komputera był nowy procesor Motorola 68030 wspomagany przez koprocesor 68882 i procesor DSP 56001. Warto tu wspomnieć, że identyczny komplet napędzał debiutujący dopiero 4 lata później Atari Falcon. Komputer posiadał 8 MB pamięci RAM, co było olbrzymią ilością. Powiedzmy tylko, że 4 MB pamięci kosztowały wówczas ok. 1500 dolarów.

Całość zamknięto w nietypowej szkieletowej obudowie wykonanej ze stopu magnezu i pomalowanej na czarny kolor. W komplecie otrzymywaliśmy też czarny monitor o olbrzymiej rozdzielczości 1120 x 832

pikseli. Pamiętajmy, że przeciętny PC oparty o procesor 8088 lub 286 oferował ekran rzędu zaledwie 640 x 480 punktów. Komputer NeXT miał także wbudowany Ethernet. Niemniej rewolucyjny był dostarczany system operacyjny. Oparty na uniksowym jądrze Mach z graficznym GUI system o nazwie NeXTSTEP wprowadzał nowe spojrzenie na nowoczesny system operacyjny. Dzisiejszy MacOS X to w prostej linii następca NeXTSTEPa.

Sprzedaż pierwszych egzemplarzy rozpoczęła się w zasadzie w 1989 roku (głównie do ośrodków akademickich), a na masowy rynek trafiły w cenie 9999 dolarów, ale dopiero w





1990 roku. Steve Jobs zapytany o to, czy nie jest rozczarowany opóźnieniami odpowiedział, że ten sprzęt wyprzedza swój czas, więc nie ma się czym martwić.

W 1991 roku niezbyt sprawny napęd magneto-optyczny został zastąpiony dyskiem twardy, które w końcu stały się tańsze i stacją dyskietek. Wymieniono również procesor na model Motorola 68040. Wprowadzono dodatkową linię sprzętową o nazwie "Light", odchudzonych i tańszych w porównaniu do "pełnego" modelu Cube. W 1992 firma NeXT rozpoczęła pracę nad dwoma projektami: portem NeXTSTEP na platformę Intel'a oraz - wraz z Apple, IBM i Motorolą - nad zastąpieniem procesorów 68k nową serią PowerPC. W 1993 roku NeXTSTEP dla komputerów z procesorami Intel'a był już gotowy. Jednocześnie zrezygnowano z prac nad komputerem opartym na PowerPC, zamykając produkcję i

sprzedaż sprzętu. Firma przemianowana została na NeXT Software. Ogółem sprzedano ok. 50 tys. sztuk komputerów NeXT. Warto wspomnieć, że w ankiecie przeprowadzonej w USA z okazji roku 2000, komputery NeXT zajęły jedno z czołowych miejsc na liście dwudziestowiecznych

przedmiotów kolekcjonerskich, w które warto zainwestować. Rzeczywiście, dzisiejsza wartość tych maszyn przewyższa wielokrotnie wartość innych antycznych komputerów - na przykład "zwykłych" Apple o podobnych parametrach sprzętowych.

W latach 1993 - 1996 NeXT Software opracowała porty NeXTSTEP dla komputerów PA-RISC oraz SPARC, a także kontynuację systemu, zwaną OPENSTEP, która uniezależniała się od jądra o nazwie Mach. W 1997 roku NeXT został wykupiony przez firmę Apple za 400 milionów dolarów, która poszukiwała systemu operacyjnego mogącego zastąpić klasyczny MacOS. Steve Jobs został "konsultantem" Apple, a niedługo później ponownie prezesem. Powszechnie uważa się, że był to krok, który postawił Apple ponownie na nogi po kilku niezbyt przychylnych latach.

Na koniec jeszcze jedna ciekawostka. Właśnie na komputerze NeXT w 1991 roku Tim Berners Lee pracując dla CERN opracował koncepcję hypertextu i napisał pierwszy serwer sieciowy oraz pierwszą sieciową przeglądarkę. Tak więc w ten sposób narodziło się WWW, bez którego dzisiaj nie wyobrażamy sobie pracy w sieci.

Lukasz Nowak

Specyfikacja techniczna komputera NeXTstation:

Procesor:	Motorola 68030 z zegarem 25 MHz
Koprocesor:	68882 i 56001
Pamięć RAM:	8 MB (maks. 32 MB)
Grafika:	1120 x 832 pikseli
Kolory:	2-bitowy (czarny, biały i dwa odcienie szarości)
Dźwięk:	koprocesor Motorola 56001/25 MHz - 16-bitowy dźwięk 44,1 KHz, stereo, 24 KB RAM rozszerzalne do 576 KB
Porty I/O:	SCSI wewnętrzne, SCSI2 zewnętrzne, DSP, wideo, NeXT Laser Printer, 2 porty RS232, Ethernet
Stacja FDD:	o pojemności 2,88 MB
Dysk twardy:	o pojemności od 100 MB - 1,5 GB
System:	NeXTSTEP, OPENSTEP

Analogowy zapis dźwięku

Co oznacza termin "technika analogowa"? Mówiąc najprościej, obróbkę sygnałów w ich podstawowej niezmiennionej, ciągłej postaci, czyli w ich naturalnym widmie częstotliwościowym.

Ciągła postać oznacza w praktyce, że jeśli zakres zmienności sygnału wynosi od 0 do 1, to jego amplituda może w dowolnej chwili przyjąć dowolną wartość z tego przedziału i jest określona w całym okresie trwania sygnału. Dokładność określenia chwilowej wartości sygnału jest ograniczona w zasadzie jedynie dokładnością stosowanych przyrządów pomiarowych i warunkami pomiaru.

Zapis analogowy oznacza, że sygnał jest rejestrowany na nośniku właśnie w naturalnej, ciągłej postaci. Jedyne zabieg, jakiemu sygnał jest poddany to ewentualnie modulacja, umożliwiającą trwały zapis.

Klasyczny przypadek zapisu w technice analogowej to przykładowo zapis dźwięku na taśmie magnetofonowej Compact Cassette, lub obrazu na taśmie magnetowidu VHS. Taśma magnetyczna przesuwana przed głowicą zapisującą. Głowica wytwarza zmienne pole magnetyczne, dokładnie odwzorowujące przebieg zapisywanego sygnału. Dzięki oddziaływaniu pola na taśmę, sygnał ana-

logowy zostaje w niej odwzorowany w postaci tak zwanej pozostałości magnetycznej, czyli lokalnych zmian namagnesowania nośnika.

Zarejestrowany w ten sposób sygnał ma przebieg dokładnie odzwierciedlający przebieg źródła, jednak jest obciążony poważnymi problemami jakościowymi.

Wszelkie szумы, przydźwięki i zakłócenia, jakie powstają w układach elektronicznych toru zapisu oraz w połączeniach kablowych sumują się z sygnałem użytecznym, zniekształcając jego przebieg i obniżając jakość późniejszego odtwarzania. Wraz z kolejnymi cyklami odczytu, wskutek bezpośredniego kontaktu głowicy z nośnikiem, stopniowemu zniszczeniu ulega warstwa ferromagnetyczna przechowująca pozostałość

magnetyczną, a tym samym spada wierność nagrania, zanikają jego szczegóły. Technikę analogową można więc podsumować następująco:

- olbrzymią zaletą (niemożliwą do uzyskania w technikach cyfrowych) jest ciągłe odwzorowanie sygnału oraz przetwarzanie go w naturalnej postaci,

- zasadniczą wadą jest trudność w eliminacji zakłóceń i szumów, której skutkiem jest słaba dynamika sygnału oraz niska wartość stosunku sygnał/szum.

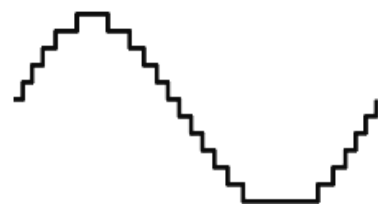
W technice cyfrowej sygnał przetwarzany jest z postaci naturalnej, ciągłej, do reprezentacji numerycznej, czyli ciągu dyskretnych wartości liczbowych.



Dźwięk oryginalny



Zapis analogowy



Zapis cyfrowy

Przetwarzanie analogowo-cyfrowe (A/C) składa się z trzech podstawowych procesów:

- próbkowania,
- kwantyzacji,
- kodowania.

PRÓBKOWANIE

Próbkowanie polega na określeniu wartości sygnału ciągłego w określonych odstępach czasu. "Chwile próbkowania" są określone przez częstotliwość jeden z podstawowych parametrów przetwarzania A/C.

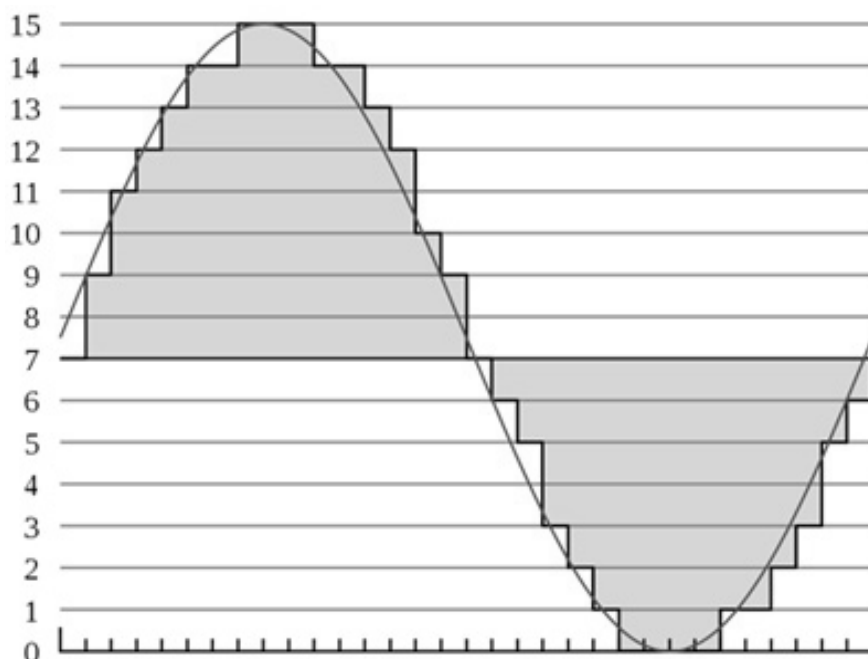
Wskutek tego procesu, zamiast przebiegu ciągłego (czyli analogowego), określonego w całym przedziale czasowym, uzyskujemy zbiór dyskretnych wartości, które można przedstawić jako tak zwany przebieg schodkowy, ponieważ w okresach czasu dzielących pobranie kolejnych próbek jego wartość jest stała.

Oczywiste jest, że w miarę wzrostu częstotliwości próbkowania, wynikowy przebieg schodkowy coraz wierniej przybliża kształt przebiegu analogowego. Zgodnie z teorią przetwarzania sygnałów, minimalna częstotliwość próbkowania musi być dwukrotnie wyższa od granicznej częstotliwości przetwarzanego sygnału.

KWANTYZACJA

W tym kroku, wartości sygnału uzyskane drogą próbkowania, należące nadal do całego zakresu zmienności sygnału, zostają "zaokrąglane" w taki sposób, by można je było przedstawić przy pomocy skończonej liczby wartości, wynikającej z rozdzielczości przetwarzania. Mówiąc w dużym uproszczeniu, na przykład wartości z przedziału 0-0,1 zostają określone jako 0,1, 0,1 -0,2 jako 0,2 itd.

W związku z tym powstaje tak zwany błąd kwantyzacji, wynikający z tego, że reprezentując ciągle zakres zmienności sygnału przy pomocy kilku



wartości dyskretnych, tracimy bezpowrotnie informację o małych zmianach w obrębie przedziałów pomiędzy sąsiednimi wartościami. Błąd ten jest tym większy im mniejsza jest liczba przedziałów kwantyzacji, a więc mniejsza rozdzielczość.

KODOWANIE

Na tym etapie liczbowe kody dyskretnych wartości, do jakich został sprowadzony sygnał źródłowy, zostają zapisane w postaci liczbowej, czyli w przypadku binarnej techniki cyfrowej, w formie liczb zapisanych w systemie dwójkowym - ciągu zer i jedynek.

Przetwarzanie cyfrowo-analogowe, z którym mamy do czynienia przy odtwarzaniu sygnału, polega na przetworzeniu ciągu liczb na przebieg schodkowy, a następnie na filtracji wygładzającej tak, by przybliżyć początkowy sygnał analogowy. Uzyskany w ten sposób sygnał cyfrowy jest zapisywany na nośniku.

Zamiast sygnału analogowego, urządzenia rejestrują ciąg 0 i 1, a więc w zamian nieskończonej liczby amplitud sygnału analogowego, uzyskujemy tylko dwie dyskretne wartości.

Dzięki temu, że zapisujemy, a następnie odczytujemy jedynie dwa stany logiczne, możliwe jest skuteczne zabezpieczenie się przed wpływem zakłóceń oraz zniekształceń.

Jeśli przyjmiemy, że logiczne "0" zapisujemy jako amplitudę -0,5V, a logiczne "1" jako +0,5V, łatwo możemy sobie wyobrazić, iż szумы i zakłócenia, nawet na poziomie 0,1V (czyli 10%) nie wpłyną na prawidłowość dekodowania tak znacznie odbiegających od siebie poziomów.

Podobnie, zużycie lub częściowe rozmagnesowanie taśmy, na której sygnał został zapisany, sprawi, że zamiast +/-0,5V zostaną odczytane poziomy niższe (+/-0,3V), ale nadal prawidłowo będzie zdekodowany ciąg 0 i 1, czyli prawidłowa postać sygnału.

W celu dalszej eliminacji zakłóceń, w zapisie cyfrowym są stosowane przeróżne mechanizmy zabezpieczeń, takie jak suma kontrolna, przeplot, kodowanie blokowe i kanałowe, umożliwiające nawet rekonstrukcję zniekształconych danych.

Opracował: Kamil Stokowski



Grafika wielokolorowa

Chciałbym przybliżyć Czytelnikom naszego pisma pakiet Bifrost Engine, który udostępnia obsługę grafiki tworząc zestaw o wymiarach 9x9 składający się z animowanych wielobarwnych płytek - 16x16 pikseli, bez uciążliwego migotania.

Początkowe wersje tego narzędzia były oparte na silniku ZXodus, jednak później kod został przepisany. ZXodus pozwolił stworzyć programy na Spectrum 48K przy zastosowaniu automatycznie generowanej mapy na ekranie. Bifrost działa w podobny sposób, ale udostępnia również dodatkowe funkcje takie jak animowane kafelki.

W TEORII

Technicznie pakiet jest wystarczająco elastyczny, aby wspierać tworzenie programów nieco podobnych do rewolucyjnej gry Buzzsaw autorstwa Jasona Railtona. Może łączyć statyczne i animowane wielobarwne elementy i umożliwia programistom bezpośrednie manipulowanie wielobarwnymi obszarami na ekranie w celu stworzenia dodatkowych efektów specjalnych.

Bifrost pozwala używać jednocześnie statycznych i animowanych wielokolorowych elementów, a także bezpośredniego manipulować różnymi

obszarami ekranu. Ruch jest płynny nawet przy uruchamianiu programów w Basicu.

Wyświetlanie wymaga wykonania wielu operacji przetwarzania, co sprawia, że program jest wolniejszy, ale jednocześnie jest szybszy o ok. 75% niż przy wykorzystaniu silnika ZXodus. Można więc powiedzieć, że program jest średnio 2 razy szybszy niż bez użycia Bifrost.

Cała mapa zostaje zaktualizowana, aby wyświetlić ekran około 4 razy na sekundę (dokładniej 6 elementów na ramkę, a więc 3,7 razy na sekundę). Oznacza to, że każda zmiana dokon-

ana na mapie zajmuje tylko średnio 135 ms na ekranie. Jest to wynik dwa razy lepszy niż dla silnika ZXodus.

W Bifrost można łatwo zmieniać ustawienia, aby wybrać odpowiedni rozmiar animacji i szybkość (2 lub 4 klatki na sekundę) dla każdego animowanego elementu.

Silnik jest kompatybilny z ZXodus, przyjmuje ten sam format obrazów i mechanizm mapowania ekranu. Dzięki temu łatwiej przenieść się na nowy silnik, gdy okaże się, że nasz program wymaga konkretnej szczególnej funkcji do uzyskania określonych rezultatów.





Bieżąca wersja Bifrost obsługuje wszystkie standardowe modele Spectrum, wliczając w to 48K, 128K, +2, +2A oraz +3.

W PRAKTYCE

Najpierw należy wyczyścić obszar pamięci przy użyciu instrukcji CLEAR, a następnie załadować obrazy poszczególnych elementów i Bifrost. Na przykład:

```
CLEAR 48499
LOAD "TILES" CODE
LOAD "BIFROST*" CODE
```

Potem nie ma potrzeby inicjowania czegokolwiek. Wystarczy włączyć wielokolorowe renderowanie, używając następującej linii:

```
RANDOMIZE NOT USR
64995
```

lub wyłączyć ten mechanizm używając:

```
RANDOMIZE NOT USR
65012
```

Silnik używa mapy elementów 9x9, czyli razem 81 pozycji, zaczynając od adresu 65281. Gdy wykonamy instrukcję POKE na ekranie pojawi się odpowiedni kolorowy obraz. Na przykład:

```
POKE 65281,0
POKE 65281+80,5
```

Powyższe linie spowodują, że w lewym górnym rogu wielokolorowego obszaru wyświetlania pojawi się element nr 0, a element nr 5 będzie widoczny w prawym dolnym rogu. Domyślnie grafiki są stale animowane w grupach 4. Pierwsza grupa animacji składa się z elementów o numerach od 0 do 3, natomiast druga grupa - od 4 do 7 i tak dalej.

Dokładnie co 0,27 sekundy mapa jest automatycznie aktualizowana przy uwzględnieniu następnego numeru. W powyższym przykładzie pozycja



65281 zmieni się na 1,2,3,0 i będzie powtarzana w nieskończoność. Jednocześnie pozycja 65281 + 80 zostanie zmieniona na 6,7,4,5 i tak dalej.

Indeksy od 0 do 127 oznaczają animowane elementy. Jeśli chcemy wyświetlać określony obraz bez animacji, do tego numeru trzeba dodać wartość 128. Można więc utworzyć przykładowo elementy 0 i 5, które pozostaną na stałe na ekranie. Robimy to w następujący sposób:

```
POKE 65281,128+0
POKE 65281+80,128+5
```

Jest jednak wyjątek od tej reguły - jeśli pozycja mapy zawiera specjalny indeks 255, na tej pozycji nie będzie mapowany żaden element, chociaż zawartość ekranu będzie nadal wyświetlana w trybie wielokolorowym. Oznacza to, że możemy umieścić wartość 255 w dowolnej pozycji na mapie, a następnie po prostu zmienić atrybuty, tak jak poniżej:

```
POKE 65281,255
POKE 65281+80,255
```

Podczas pisania programu może wydawać się mylące manipulowanie położeniami elementów, które nieus-

tannie zmieniają się z powodu ciągłej animacji. Jednak jest to całkiem proste. Jeśli pozycja zawiera animowany element, należy ją podzielić przez 4 i obserwować efekt zmiany ramki animacji:

```
LET a=PEEK 65281
IF a<128 THEN LET
a=INT (a/4)
```

Kod Bifrost kończy się pod adresem 65280, a mapa znajduje się pod adresem od 65281 do 65361. Każdy element zajmuje 64 bajty i przechowuje 32 bajty mapy bitowej (16 par od góry do dołu), a następnie 32 bajty atrybutów przechowywanych w tej samej kolejności.

Po wykonaniu instrukcji POKE z wartością 255 możemy normalnie modyfikować informacje na mapie bitowej. Jednak bezpośrednia próba zmiany atrybutów ekranu nie będzie działać, ponieważ są one zastępowane kolorem, dlatego też trzeba zmienić atrybuty trybu wielokolorowego.

Aby sprawdzić dokładny adres pamięci należy pamiętać, że wielokolorowy obraz jest generowany z wierszy od 1 do 18, a więc są to następujące linie pikseli:



od: $8 + 1 * 8 = 16$
do: $8 + 18 * 8 + 7 = 159$

Liczby te wynikają z faktu, że współrzędne pikseli liczone są począwszy od 8 pikseli nad obszarem wyświetlania oraz w kolumnach trybu znakowego od 1 do 18.

Dalej należy zastosować kolejne obliczenia:

```
POKE 59075 + ((linia-16)*41 + delta, atrybut
```

Wartość "delta" powyżej zależy od kolumny, którą należy zmieniać zgodnie z tabelą numer 1, którą przedstawiam niżej. Przykładowo, jeśli chcemy zmienić wszystkie 8 atrybutów odpowiadających lokalizacji:

```
PRINT AT 10,9
```

możemy wykonać dłuższy kod - patrz listing nr 1.

```
POKE 65281+40,255
LET delta=17
LET rzad=10
FOR f=8+rzad*8 to 8+rzad*8+7
POKE 59075+(f-16)*41+delta, atrybut
NEXT f
```

Listing nr 1.

Jeśli chcemy zmieniać obszar wielokolorowy bezpośrednio oraz wszystkie wyniki mają pojawiać się za jednym zamachem, można użyć instrukcji HALT tuż przed wprowadzeniem zmian.

Przedstawione rozwiązanie przyjmuje, że w naszym programie używamy kafelków, dzieląc ekran na równe części - tak jak w grach. Jeżeli nie chcemy tego robić, nadal istnieje możliwość wykorzystania funkcji silnika. Najłatwiejszym sposobem jest trwałe wypełnienie obszaru mapy płytek ze specjalnym indeksem 255.

W tym przypadku Bifrost nie będzie rysował żadnych elementów, a po prostu będzie czekać, aż wiązka elektronów dotrze do obszaru ekranu, gdzie ma zacząć się wyświetlanie w trybie wielokolorowym. Dzięki temu nie musimy ładować żadnych grafik do pamięci i program może wykorzystać ten obszar do innych celów.

Bardziej złożoną alternatywą jest

opracowanie własnego kodu w Asemblerze w celu zastąpienia części mapowania ekranu. Technicznie oznacza to, że możemy zastąpić sekcję pomiędzy poniższymi adresami:

od 59031 do 59046.

Ten sposób nie jest jednak zalecany, ponieważ trudno jest napisać skompilowany kod, gdzie każda część zajmuje dokładnie ten sam czas, a jest to wymagane do wyświetlania trybu wielokolorowego bez przekłamań. Dlatego lepiej używać wielokolorowych kafelek, aby Bifrost zajął się wszystkimi kwestiami zupełnie automatycznie.

Jeśli mimo to chcemy zastosować własny kod, proponuję przeanalizować poniższy przykład:

```
org 59031
ld bc,538 ;10 T
loop:
dec bc ;538*6 T
ld a,b ;538*4 T
or c ;538*4 T
jr nz,loop ;537*12+7 T
ret nz ;5 T
jp 59047 ;10 T
```

Pokazuje on, jak zastąpić oryginalne odwzorowanie nowym kodem. Zajmuje dokładnie tyle czasu, ile trzeba, chociaż nie robi nic użytecznego. Na tej bazie można jednak rozbudować własny algorytm.

Opracował: Łukasz Nowak

Numer kolumny

Wartość "delta"

1	4
2	5
3	7
4	8
5	10
6	11
7	14
8	15
9	17
10	18
11	32
12	33
13	28
14	29
15	24
16	25
17	20
18	21

Tabela nr 1.



Dźwięk bez komplikacji

Każdy wie, że ZX Spectrum posiada wbudowany głośnik - przynajmniej mam taką nadzieję. Skorzystanie ze skromnych możliwości dźwiękowych w interesujący sposób to zupełnie inna sprawa. Na szczęście nie jest to bardzo skomplikowane, co zaraz pokażę.

Głośnikiem w Basicu steruje się przy pomocy polecenia BEEP, któremu trzeba podać czas trwania oraz wysokość dźwięku. Obie wartości są wartościami liczbowymi. Czas trwania jest podawany w sekundach, a wysokość w półtonach ponad środkowym dźwiękiem C. Nuty poniżej C uzyskuje się za pomocą wartości ujemnych. Aby otrzymać wyższe lub niższe nuty, musimy dodawać lub odejmować 12 dla każdej oktawy. Przykładowy program widać w ramce obok.

Założmy, że melodia jest zapisana w tonacji c-moll. Wartości nut można zapisać na przykład tak:



Dzięki takiemu zapisowi można dość łatwo znaleźć wartość określonej nuty na pięciolinii.

```
10 BEEP 1,0: BEEP 1,2: BEEP .5,3
20 BEEP .5,2: BEEP 1,0
30 BEEP 1,0: BEEP 1,2: BEEP .5,3
40 BEEP .5,2: BEEP 1,0
50 BEEP 1,3: BEEP 1,5: BEEP 2,7
60 BEEP 1,3: BEEP 1,5: BEEP 2,7
70 BEEP .75,7: BEEP .25,8: BEEP .5,7
80 BEEP .5,5: BEEP .5,3: BEEP .5,2
90 BEEP 1,0
100 BEEP .75,7: BEEP .25,8
110 BEEP .5,5
120 BEEP .5,3: BEEP .5,2: BEEP 1,0
130 BEEP 1,0: BEEP 1,-5: BEEP 2,0
140 BEEP 1,0: BEEP 1,-5: BEEP 2,0
```

Listing nr 1.

Jeśli chcemy zmienić tonację danego utworu, najlepiej utworzyć zmienną i dodawać ją do każdej wartości nuty. W naszym przykładzie może to wyglądać następująco:

```
BEEP 1,t+0
BEEP 1,t+2
BEEP .5,t+3
```

Zanim uruchomimy program trzeba oczywiście nadać odpowiednią wartość zmiennej (tutaj "t") — 0 dla tonacji C, 2 dla D, 12 dla C o oktawę wyżej, itd. Dzięki temu możemy dostroić komputer do instrumentu przez użycie wartości ułamkowych naszej zmiennej.

Musimy również wyliczyć czas trwania każdej nuty. Dla wolnego utworu można przyjąć jedną sekundę na ćwierćnutę, a resztę oprzeć na tym wyborze, czyli ósemka zajmie czas pół sekundy itd.

Można to zrobić też w bardziej elastyczny sposób, mianowicie utworzyć drugą zmienną, która będzie przechowywała długość ćwierćnuty.

```
BEEP .5,t+2
BEEP 1,t+0
```




Liczby pseudolosowe

W tym artykule omówię funkcję RND oraz polecenie RANDOMIZE. Są one wykorzystywane w związku z liczbami losowymi, i trzeba uważać, aby ich nie mylić. Z pewnością nie ułatwia tego fakt, że znajdują się na tym samym klawiszu (znak "T").

RND wykonuje obliczenia i tworzy pewien wynik, jak każda inna funkcja. Jednak nie potrzebuje argumentu. Przy każdym użyciu wynikiem jest nowa liczba losowa w zakresie od 0 do 1. Czasami może ona być równa 0, lecz nigdy nie jest równa 1. Wypróbujmy przykład:

```
10 PRINT RND
20 GO TO 10
```

Teraz można zobaczyć, jak różnią się między sobą kolejne wyniki. Czy możemy w nich rozpoznać jakąś prawidłowość? Teoretycznie nie powinno być takiej możliwości, bo wynik powinien być losowy. W rzeczywistości RND nie jest naprawdę losowe, ponieważ przyjmuje wartości ze stałego ciągu 65536 liczb. Jednakże są one tak ze sobą wymieszane, że nie tworzą żadnego oczywistego układu, a o funkcji RND mówimy, że jest pseudolosowa. Zatem RND daje liczby "losowe" w zakresie od 0 do 1. Łatwo można otrzymać także liczby w innych zakresach. Na przykład:

```
RND*5
```

daje wynik pomiędzy 0 a 5, natomiast dzięki bardziej skomplikowanej formule:

```
1.3+0.7*RND
```

otrzymamy liczby pomiędzy 1.3 a 2. Aby otrzymywać liczby całkowite, można użyć innej funkcji - INT, trzeba jednak pamiętać, że zawsze zaokrągla ona wartości "w dół". Obie funkcje można ze sobą mieszać, na przykład:

```
1+INT (RND*6)
```

Zapis RND*6 spowoduje uzyskanie liczb w zakresie od 0 do 6, lecz analogicznie - nigdy nie osiągnie wartości 6. Tak więc poniższa funkcja:

```
INT (RND*6)
```

daje wynik: 0, 1, 2, 3, 4 lub 5. Dzięki temu możemy w prosty sposób napisać program symulujący rzucaniem kostkami do gry:

```
10 CLS
20 FOR n=1 TO 2
30 PRINT 1+INT (RND*6)
; " ",
40 NEXT n
50 INPUT a$
60 GO TO 10
```

Polecenie RANDOMIZE jest wykorzystywane do zmuszania RND, aby rozpoczynało od określonego miejsca w ciągu liczb, co można zobaczyć za pomocą następnego krótkiego programu:

```
10 RANDOMIZE 1
20 FOR n=1 TO 5
30 PRINT RND,
40 NEXT n
50 PRINT: GO TO 10
```

Teraz po każdym wykonaniu RANDOMIZE 1 ciąg liczb RND rozpoczyna się od wartości:

0.0022735596

W poleceniu RANDOMIZE można oczywiście użyć innych liczb od 1 do 65535, aby RND rozpoczynało w innym miejscu "skali".

Samo RANDOMIZE bez żadnej liczby spowoduje uzyskanie przypadkowej wartości RND, co można zaobserwować po wykonaniu poniższych linii:

```
10 RANDOMIZE
20 PRINT RND
30 GO TO 10
```

Otrzymany ciąg liczb nie jest do końca losowy, gdyż RANDOMIZE wykorzystuje czas od momentu uruchomienia komputera. Lepszą losowość otrzymamy poprzez zamianę GO TO 10 poleceniem GO TO 20.



Trudna obsługa klawiatury

Klawiatura Spectrum jest podobna do typowej "maszyny do pisania", ale jednocześnie może stanowić niezły orzech do zgryzienia. Każdy klawisz może wykonywać więcej niż jedną funkcję i rzeczywiście mamy do dyspozycji wiele funkcji, które w dużej części w języku Basic muszą być wpisywane w określony sposób. Nie wystarczy uzyskać dany efekt na ekranie. Weterani tematu retro mają tę wiedzę w małym palcu, ale młodszy Czytelnicy mogą być nieco zdezorientowani, nawet jeżeli wcześniej - przed korzystaniem z prawdziwego ZX Spectrum - używali emulatorów.

Gdy Spectrum jest włączony bez wykonywania żadnych działań, ekran ukazuje napis "firmowy", czyli:

```
(C) 1982 Sinclair Research Ltd
```

Naciśnięcie dowolnego klawisza umieszcza w dolnym wierszu ekranu słowo wypisane pod literą na tym klawiszu, które nazywamy słowem kluczowym. Komputer oczekuje

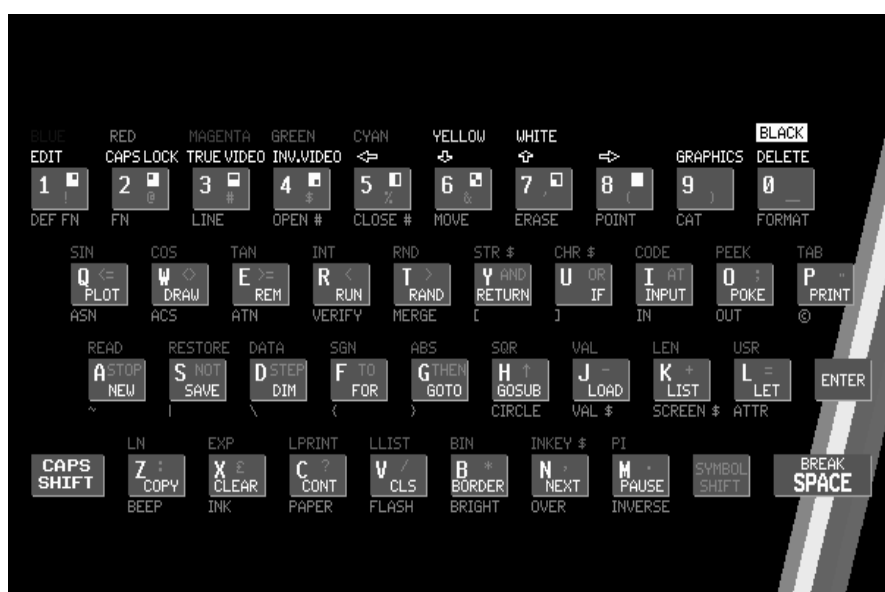
rozkazu, który mu powie, co ma robić, a wszystkie rozkazy rozpoczynają się od właśnie słów kluczowych. W przeciwieństwie do większości innych komputerów Spectrum umożliwia, a ściślej mówiąc - nakazuje wprowadzanie słów kluczowych poprzez naciśnięcie pojedynczego klawisza.

Aby pomóc ci w zorientowaniu się, w jakim trybie znajduje się klawiatura, na ekranie pojawia się litera w negatywie, biała na czarnym tle, która wskazuje pozycję, gdzie trafi kolejny znak po naciśnięciu klawisza. Jest to dość nietypowy kursor, który - w przeciwieństwie do nowoczesnych systemów - ma więcej funkcji niż tylko wskazywanie lokalizacji w treści listingu programu.

Na przykład, jeśli po włączeniu komputera naciśnięty zostanie klawisz "P", na ekranie pojawi się słowo kluczowe PRINT. Znak cudzysłowu jest zaznaczony na tym samym klawiszu.

Jednak aby go otrzymać, musimy jednocześnie nacisnąć dwa klawisze - przytrzymać klawisz SYMBOL SHIFT, który znajduje się w dolnym rzędzie klawiszy po prawej stronie klawiatury oraz literę P. Kursor zmienia się na pozycję "L", ponieważ teraz komputer oczekuje liter (ang. letter), a więc możemy wpisać dowolny tekst.

Ogólnie, wszystko pokolorowane na białą na klawiszu wymaga naciśnięcia z klawiszem CAPS SHIFT,





```
10>LET intt=4
11 CLS
20 BORDER 5
30 PAPER 3
40 INK 5
41 CLS
50 PLOT 0,0
55 DRAW 255,0
100 FOR n=1 TO 175
110 PLOT 128-88,n
120 DRAW -40,0
121 PLOT 128+88,n
122 DRAW 39,0
130 NEXT n
200 LET a=176
210 LET x=128-88
220 LET y=1
221 GO TO 1000
225 REM SUB draw subsquare
230 FOR m=0 TO INT (a/3)
231 IF yy+INT (a/3)+m>175 THEN
GO TO 260
```

K

pretuje to, co wpiszesz. Na początku wiersza będzie to mrugające "K", co oznacza oczekiwanie na "słowo kluczowe" (ang. keyword).

Słowo kluczowe jest jednym ze specjalnych słów występujących na początku wiersza i dających systemowi ogólne pojęcie na temat funkcji rozkazu. Jest to oczekiwane na początku wiersza, dlatego po naciśnięciu klawisza, na przykład "P", komputer zinterpretuje go jako instrukcję PRINT.

Gdy słowo kluczowe zostanie wprowadzone, system nie oczekuje drugiego, a więc to, co teraz wpisujemy zostanie zinterpretowane jako zwykłe litery. Aby poinformować o tym użytkownika, komputer zmienia kursor na "L". Dlatego warto obserwować różne tryby kursora.

a wszystko oznaczone na czerwono wymaga naciśnięcia z klawiszem SYMBOL SHIFT.

Trochę to skomplikowane, a przecież mamy do czynienia ze "zwykłym" ZX Spectrum. Rozkaz rozpoczynający się słowem PRINT nakazuje komputerowi wypisać na ekranie litery ujęte w cudzysłów. Aby wpisany rozkaz został wykonany przez komputer, należy użyć klawisza ENTER, czyli nie ma tu kolejnej niespodzianki. Migający znak zapytania oznacza błąd, który trzeba zlokalizować samodzielnie, bowiem Basic nie pomaga w tej kwestii.

Zwróćmy uwagę na ważną cechę, która dzisiaj jest raczej pomijana. Mianowicie litera "O" oraz cyfra 0 są przedstawiane przez różne znaki. Warto to zapamiętać. Cyfra 0 zawsze jest przekreślona kreską, co pomaga rozróżnić dwa podobne kształty na ekranie. Komputer zawsze zinterpretuje literę "O" jako tekst i trzeba na to uważać. Podobnie cyfra 1 oraz mała litera "l" różnią się i nie można używać ich zamiennie. Tryb klawiatury w Basicu jest bardzo ważny, dlatego podsumujmy dotychczasowe

```
5 CLS
10 FOR n=0 TO 7
20>NEXT n
```

```
30 DATA BIN 11001111,BIN 1100111
1,BIN 00000000,BIN 00110000,BIN
00110000,BIN 00000000,BIN 110011
11
```

informacje. Migający znak "L" wskazuje miejsce na ekranie, gdzie komputer będzie wstawiał tekst. Nie zawsze jest to "L", bowiem gdy włączymy komputer i naciśniemy ENTER, tekst zmieni się w kursor z oznaczeniem "K". Widoczna litera mówi, w jaki sposób komputer zinter-

Jeśli chcemy wpisać dużą ilość WIELKICH liter bez ciągłego naciskania klawisza CAPS SHIFT, możemy sprawić, że wszystkie litery będą pojawiać się jako duże. Trzeba tylko nacisnąć CAPS LOCK, co jest w zasadzie oczywiste. Nie piszę tego jednak bez celu, bo w tym momencie



odkrywamy kolejny tryb pracy Basica. W celu jego zaznaczenia kursor zostanie zastąpiony przez migające "C" (ang. capitals).

Jeżeli naciśniemy CAPS LOCK podczas trybu "K", nie będzie widać różnicy, ale efekt stanie się widoczny po wprowadzeniu słowa kluczowego, gdy komputer będzie w trybie "C", zamiast trybu "L".

Oprócz słów kluczowych, liter, cyfr i różnych wyrażeń naukowych, do dyspozycji mamy także osiem znaków graficznych. Pojawiają się one na klawiszach od 1 do 8 i mogą być wyświetlane na ekranie w podobny sposób jak litery i cyfry. Aby to zrobić klawiatura musi zostać przełączona w tryb graficzny. Dokonuje się tego przez naciśnięcie klawiszy CAPS SHIFT i 9. Kursor zmieni się teraz na oznaczenie "G". Ponowne naciśnięcie klawisza 9 przywróci z powrotem tryb "L".

Ostatni tryb, na który może być przełączona klawiatura to tak zwany tryb rozszerzony oznaczony literą "E" (ang. extended). Otrzymujemy go poprzez równoczesne naciśnięcie

```

4 REM BY UAXALON - 1995
5>GO SUB 200
10 LET w=CODE INKEY$
20 IF w=0 THEN GO TO 10
25 PRINT CHR$ w;
30 IF w>=48 AND w<=57 THEN LET
w=w-48: GO TO 60
40 IF w>=65 AND w<=91 THEN LET
w=w-55: GO TO 60
50 IF w>=97 AND w<=123 THEN LE
T w=w-87: GO TO 60
55 GO TO 10
60 LET s=m(w+1)
70 IF s=0 THEN GO TO 10
80 LET c=s-(INT (s/10)*10)
90 IF c=1 THEN BEEP .2,0: GO T
O 110
100 IF c=2 THEN BEEP .6,0
110 LET s=INT (s/10): FOR x=1 T
O 12: NEXT x
120 GO TO 70
200 DIM m(36)

```

RUN █

klawiszy CAPS SHIFT i SYMBOL SHIFT. Pozwala to wykorzystywać większość funkcji naukowych. Ponowne naciśnięcie tych samych dwóch klawiszy przełączy klawiaturę z powrotem na tryb "L".

Aby poprawić swój nieprawidłowo wpisany wiersz, naciskamy CAPS

SHIFT i 5, aż kursor ustawi się tuż za znakiem, który został wstawiony omyłkowo. Dalej naciskamy DELETE (czyli CAPS SHIFT i 0), aby usunąć błędny znak, a następnie wpisujemy prawidłowy. Nowa litera zostanie wstawiona bez nadpisywania innych znaków.

Mam nadzieję, że po przeczytaniu tego artykułu nie jesteście jeszcze bardziej zdezorientowani, niż na początku. Klawiatura Spectrum i możliwe do wykorzystania tryby pracy wydają się proste w użyciu, ale praktyka wygląda trochę inaczej. Dzisiaj jesteśmy przyzwyczajeni raczej do jednego lub dwóch trybów kursora, który nie posiada dodatkowych oznaczeń.

W przeszłości bywało inaczej, a produkty firmy Sinclair są charakterystycznym przykładem. Początkowo możecie mieć problem z "opanowaniem kursora", ale w miarę korzystania z różnych funkcji Basica można dojść do wprawy. Osobiście robiłem to będąc uczniem szkoły podstawowej, więc tym bardziej może to się udać starszym kolegom po latach. Potrzeba tylko trochę wytrwałości.

```

! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _
` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~ ¢
. . . . . A B C D E F G H I J K L M N O P
Q R S T U R N D I N K E Y $ P I F N P O I N T S C R E E N $
A T T R A T T A B V A L $ C O D E V A L L E N $
I N C O S T A N A S N A C S A T N L N E X P I N
T S Q R S G N A B S P E E K I N U S R S T R $ C
H R $ N O T B I N O R A N D < = > = < > L I N E T
H E N T O S T E P D E F F N C A T F O R M A T M O
V E E R A S E O P E N # C L O S E # M E R G E V E
R I F Y B E E P C I R C L E I N K P A P E R F L A S H
B R I G H T I N V E R S E O V E R O U T L P R I N T
L L I S T S T O P R E A D D A T A R E S T O R E N E W
B O R D E R C O N T I N U E D I M R E M F O R G O
T O G O S U B I N P U T L O A D L I S T L E T P A
U S E N E X T P O K E P R I N T P L O T R U N S A V
E R A N D O M I Z E I F C L S D R A W C L E A R R E
T U R N C O P Y

```

0 OK, 30:1

CPC 464 i grafika

We wczesnych latach osiemdziesiątych ZX Spectrum i Commodore 64 walczyły o panowanie nad światem w grupie komputerów domowych. W 1984 r. dołączyła do nich kolejna 8-bitowa maszyna o nazwie Amstrad CPC. Choć pojawiła się dość późno, okazała się niezwykle popularna. Jednym z powodów był fakt, że wraz z komputerem dostarczono 12-calowy monitor (kolorowy lub zielony), co oznaczało, że nie musimy walczyć o korzystanie z rodzinnego telewizora.

Pierwszym modelem był CPC464. Ta maszyna miała całkiem dużą moc obliczeniową. Głównym procesorem był procesor Zilog Z80A pracujący z szybkością 3,3 MHz oraz 64 KB pamięci RAM. Dźwięk był generowany przez układ AY-3-8912 firmy General Instruments, który pozwalał uzyskać trzy kanały z zakresem ośmiu oktaw. Dodano do nich dodatkowy kanał szumu wykorzystywany do efektów w grach.

Po CPC464 pojawił się CPC664, który miał wbudowaną 3-calową stację dyskietek. Niedługo później

Amstrad wydał model CPC6128 przeznaczony do bardziej poważnych zastosowań. W tym modelu pamięć została podwojona (do 128 KB), ale procesor Z80 mógł adresować tylko 64 KB na raz. Dlatego programiści musieli przełączać się między dwoma blokami, aby skorzystać z dodatkowej pamięci.

Amstrad wyprodukował później modele CPC464 plus, CPC6128 plus i konsolę gier GX4000. Te maszyny miały ulepszone możliwości graficzne, ale niestety zostały wydane zbyt późno, aby osiągnąć duży sukces. W tym czasie 8-bitowe maszyny traciły popularność na korzyść 16-bitowych maszyn, takich jak Atari ST i Amiga.

Na podstawowym sprzęcie CPC mamy dostępną paletę 27 różnych kolorów, które są tworzone poprzez zmieszanie trzech składowych: czerwieni, zieleni i niebieskiego. Nie wszystkie kolory można używać jednocześnie, ale tylko 16. Jeśli chcemy używać wyższej rozdzielczości, ilość dostępnych kolorów spada.

Oto obsługiwane tryby ekranu:

- **Mode 0** - niska rozdzielczość - 160 x 200 - 16 kolorów,
- **Mode 1** - średnia rozdzielczość - 320 x 200 - 4 kolory,
- **Mode 2** - wysoka rozdzielczość - 640 x 200 - 2 kolory,
- **Mode 3** - tryb nieoficjalny - 160 x 200 - 4 kolory.





Wszystkie wymienione tryby można rozszerzyć używając funkcji Over-scan, która jest również znana z Amigi. Na przykład, możliwe jest rozszerzenie rozdzielczości typu Mode 0 do 192 x 264 w 16 kolorach.

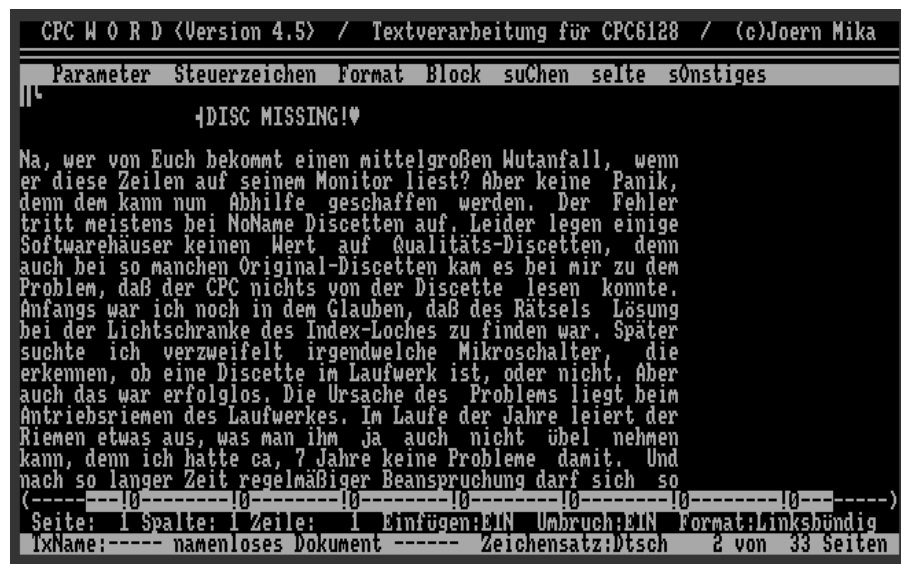
Mode 0 jest trybem najbardziej popularnym w CPC. Ze względu na niską rozdzielczość i ograniczenia palety, łatwo można zauważyć niedogodności w postaci braku antyaliasingu i ditheringu.

W modelu CPC plus tryby ekranu są takie same, ale paleta składa się z 4096 kolorów, zamiast 27 - jak wcześniej. Podobnie jak w podstawowym CPC, jednocześnie można



bardziej rozbudowany. Pozwala łączyć tekst i grafikę w sposób przypominający nowoczesne edytory. Za

jego pomocą można z powodzeniem tworzyć dokumenty w formacie A4. Pro-Design obsługuje myszkę, umożliwiając stosować różne czcionki, posiada również sterowniki do drukarek.



Dzisiaj dla CPC istnieje system Sym-BOS posiadający graficzny interfejs użytkownika i pozwalający pracować w trybie wielozadaniowym. Doskonale wykorzystuje on specyficzne cechy komputerów Amstrada, w tym charakterystyczną paletę barw oraz urządzenia peryferyjne. Ostatnia wersja pochodzi z 2016 roku. Komputery CPC dobitnie pokazują jak szeroko można wykorzystać potencjał sprzętu 8-bitowego. W ostatnim czasie pojawiają się również rozszerzenia sprzętowe, w tym karta graficzna, o której niebawem napiszemy szerzej.

używać tylko 16 z 4096 kolorów, ale oczywiście istnieją różne demo, które łamią to ograniczenie. Jedną z najbardziej znanych gier w sieci CPC jest "Sorcery Plus" wydana w 1985 roku. Programiści tej gry użyli jednocześnie dwóch z dostępnych trybów graficznych. Co ciekawe, firma Amstrad używała tej produkcji do promowania swojego komputera.

Ze względu na ograniczone zasoby CPC, tworzenie grafiki nie jest zbyt wygodne. Polecam uruchomić program malarski "Art Studio". Widoczne obok zrzuty ekranowe wyraźnie pokazują, jak ograniczona jest praca przy tak niskiej rozdzielczości. Innym znanym programem graficznym dla CPC jest Pro-Design, który jest

Ogólna specyfikacja techniczna komputera Amstrad CPC 464

Procesor:	Zilog Z80 4 MHz
Pamięć RAM:	64 KB
Pamięć ROM:	32 KB
Pamięć VRAM:	16 KB
Tekst:	80 x 25 (2 kolory) 40 x 25 (4 kolory) 20 x 25 (16 kolorów)
Grafika:	640 x 200 (2 kolory) 320 x 200 (4 kolory) 160 x 200 (16 kolorów)
Dźwięk:	3 kanały generator szumów, wbudowany głośnik
System:	Locomotive Basic

Konsole GX4000

W roku 1990 firma Amstrad wprowadziła serię Plus, która pod wieloma względami przewyższała poprzednie modele komputerów, przynajmniej jeśli chodzi o fabryczną konfigurację. Wyprodukowane zostały modele 464 Plus, 6128 Plus oraz konsola do gier o oznaczeniu GX-4000, która posiadała wiele ulepszeń sprzętowych.

Większość z nich dotyczyły możliwości graficznych, zwiększono paletę barw do 4096 kolorów oraz zdolność wyświetlania sprzętowych sprajtów. Pojawił się podział ekranu na dwa osobne obszary oraz przewijanie wspomagane sprzętowo. To wszystko możliwe było już wcześniej przy użyciu układu Motorola 6845, ale wymagało specjalnych technik programistycznych. Dodano również kanał DMA dla układu dźwiękowego, ale same możliwości audio pozostały niezmienione. Ulepszono zestaw poleceń BASIC obsługujących operacje dyskowe.

Modele te niestety nie radziły sobie na rynku tak dobrze, jak założył to ich producent, ani nie przyciągały oczekiwanej uwagi firm trzecich. Technologia 8 bitów nie była już tak atrakcyjna jak wcześniej, poza tym gry na cartridge'ach były droższe niż ich odpowiedniki na kasetach. Mimo to maszyny "z Plusem" sprzedawały się dobrze w krajach takich

jak Francja, gdzie Amstrad wciąż miał duże powodzenie ze względu na popularność klasycznej serii CPC.

Sprzęt został oficjalnie przemianowany na "Amstrad 464 Plus", "Amstrad 6128 Plus" i "GX4000", a skrót "CPC" został porzucony. Jednak wielu użytkowników Amstrad do tej pory określa te komputery jako "CPC Plus" ze względu na nieodłączne podobieństwa.

GX4000 to konsola do gier oparta na modelu 6128 Plus, ale posiada kontrolera stacji dyskiety oraz klawiaturę, chociaż po pewnych przeróbkach można je dodać. Konsola była

sprzedawana z dwoma wiosłkami (ang. paddle), czyli urządzeniami posiadającymi pokrętło służące do przesuwania obiektu na ekranie - wzdłuż jednej osi.

Dzisiaj często stosowane jest określenie "pokrętło sterujące", chociaż dla mnie brzmi to podobnie jak kiedyś "drążek sterowy" dla joysticka.

Nowa konsola stanowiła próbę zdobycia przez Amstrada udziału w rynku gier video, a następnie przekucia to w dominację, podobnie jak zrobiły to Nintendo czy Sega. Seria "CPC Plus" była jedną z najlepszych wśród kom-





puterów 8-bitowych, jednak została wydana zbyt późno, aby faktycznie przełożyło się to w sukces. Braki oprogramowania, brak odpowiedniego marketingu i niezbyt dobry czas dla takiej premiery spowodowały, że przygoda Amstrada na rynku gier miała szybko się skończyć.

Jak widać różnice GX4000 w porównaniu do 464 lub 6128 Plus nie były zbyt duże. Ciekawostką jest fakt, iż przycisk Pause był mapowany jako klawisz "P" na zwykłej klawiaturze CPC. Różniły się połączenia zasilacza, nieobecne były także wejścia dla stacji dyskiek, drukarki, klawiatury, magnetofonu oraz związane z nimi sygnały, na przykład Printer Busy czy Cassette Motor.

Po włożeniu cartridge'a komputer wyświetla komunikat o prawach autorskich oraz komunikat "Ready". Jeśli przyciski Fire lub kierunki są wciśnięte na pierwszym joysticku, wyświetlane są dodatkowo oznaczenia X, Z i strzałki, tak samo jak w przypadku naciśnięcia tych przycisków w urządzeniu 464/6128 Plus lub CPC. Naciśnięcie przycisku Pause powoduje wyświetlenie "P".

Oprócz joysticków można podłączyć także kontrolery typu "light gun", ale opcja ta została opracowana przez firmy trzecie. Dlatego też istnieją tylko dwie gry korzystające z tej możliwości - "Skeet Shoot" oraz "The Enforcer". Oprogramowanie na cartridge'ach może bez problemu korzystać z dodatkowych funkcji. Nie są one blokowane sprzętowo, ale trzeba wykorzystać specjalną

Specyfikacja techniczna konsoli Amstrad GX4000:

Procesor:	Zilog Z80A z zegarem 4 MHz
Pamięć:	RAM: 64 KB VRRAM: 16 KB ROM: 32 KB
Grafika:	obsługa sprajtów, scrollingu, programowane przerwania
Rozdzielczość:	Mode 0: 160x200 pikseli (16 kolorów) Mode 1: 320x200 pikseli (4 kolory) Mode 2: 640x200 pikseli (2 kolory)
Paleta kolorów:	RGB 12-bitowa (4096 kolorów)
Ilość kolorów:	maksymalnie na ekranie: 32 kolory (16 dla tła, 15 dla sprajtów, 1 dla ramki)
Inne:	maksymalna ilość kolorów może być zwiększona w Mode 0-2 przy wykorzystaniu przerwań
Sprajty:	- 16 obiektów w wysokiej rozdzielczości, - każdy obiekt może być powiększany 2 lub 4 razy zarówno w osi X, jak i Y - każdy obiekt może być 15 kolorów
Dźwięk:	- układ AY-3-8912, 3 kanały stereo - kanał DMA
Porty:	- wyjście audio - wyjście RGB Video (8-pin DIN) - 2 gniazda dla cyfrowych kontrolerów - 1 gniazdo dla analogowego kontrolera zgodnego ze standardem IBM - gniazdo RJ-11 dla kontrolera typu "Light Gun" - złącze zewnętrznego zasilacza 5V

sekwencję o długości 17 bajtów. Dlatego każdy może tworzyć nowe programy, a oprócz tego sprzęt jest w pełni zgodny z serią CPC.

Mimo skromnej bazy gier, konsola ta jest warta uwagi, choć trudno ją obecnie zdobyć. W komplecie oryginalnie dostarczana była gra "Burnin' Rubber",

czyli bardzo przyzwoite wyścigi w stylu słynnego Lotusa. Opracowano nawet kilka wersji językowych, choć wydaje się to zbędnym wysiłkiem dla wydawcy.

Gra korzysta ze specyficznych funkcji serii "Plus", między innymi różnych trybów graficznych, podziału ekranu na części oraz kanału DMA dla uzyskania efektów dźwiękowych. Dla GX4000 opracowano tylko 30 gier, wśród których znajdziemy hity jak "Robocop 2" czy "Navy Seals".

Polecam spróbować uruchomić gry dla GX4000 na emulatorze MESS, który można znaleźć na poniższej stronie internetowej:

<https://www.mess.org/>

Kamil Stokowski



Kompresja danych

W Commodore Basicu zwykle dane zapisywane są w plikach sekwencyjnych. Zajmuje to sporą ilość miejsca i czasami przydatne może być spakowanie danych. Trzeba to jednak zrobić sprytnie, ze względu na ograniczenia sprzętowe. Mój sposób można wykorzystać nawet na komputerze PET.

Jeśli nasz program zawiera instrukcji takie jak PRINT#, a zapisywana zmienna będzie zawierać liczbę 132, w pliku umieszczonych zostanie aż 5 znaków:

- SPACJA,
- 1,
- 3,
- 2,
- RETURN.

Gdy nie korzystamy z Basica w wersji 4.0, dodatkowo zapisany zostanie znak końca linii (ang. Line Feed), a to już może powodować dalsze problemy. Dlatego na końcu najlepiej dodać znak o kodzie 13, czyli na przykład tak:

```
PRINT#1, A;CHR$(13);
```

Jest to bardzo dobre rozwiązanie dla wielu celów, bowiem tak zapisane

dane można odczytać za pomocą instrukcji INPUT# i zostaną one wprowadzone tak, jakby użytkownik wpisał je z klawiatury. Cały plik składa się z konwencjonalnych znaków ASCII, które mogą być wykorzystywane w edytorach tekstu czy choćby podczas transmisji danych.

Jednak co zrobić, gdy chcemy napisać program, który będzie miał za zadanie sortowanie informacji? W takim wypadku dane powinny być zapisane inaczej, na przykład "w kolumnach", aby można było je łatwo identyfikować.

Taki plik będzie dużo większy i przy użyciu większej ilości danych, objętość staje się znacznie bardziej poważnym problemem.

Poszczególne pola rekordu możemy zapisać jako pojedyncze elementy. Najprościej można to zrobić za

pomocą kilku linii PRINT#, na przykład tak:

```
PRINT #1, A$  
PRINT #1, B$  
PRINT #1, C$  
PRINT #1, D
```

Taki układ jest łatwy do obsługi, a w razie potrzeby plik można odczytać "awaryjnie" na wiele sposobów. Nie tracimy więc danych.

Zwykle każda z informacji będzie miała ustalony maksymalny rozmiar, co oznacza konkretną ilość znaków przeznaczoną na jedną zmienną. Jak widać, zapis jest łatwy do zrozumienia, ale teraz musimy znaleźć sposób na zapisane informacji w formie skompresowanej.

Każde pole z informacją musi być taki sam rozmiar. Jeśli dane będą za długie, trzeba je skrócić przy użyciu funkcji LEFT\$, na przykład:





LEFT\$(A\$, 10)

Z kolei, gdy ciąg tekstowy będzie zbyt krótki, trzeba go rozszerzyć dodając na końcu znaki Spacji, czyli użyć schematu podobnego do poniższego:

LEFT\$(A\$+" ", 10)

W tym miejscu warto dodać uwagę. Musimy mieć pewność, że gotowy wpis będzie zawierać wystarczającą ilość znaków. Dlatego najlepiej zdefiniować dodatkową zmienną zawierającą większą ilość Spacji. Ciąg i tak zostanie "przycięty" do naszego rozmiaru.

Kolejne znaki zapisanych informacji muszą sobie odpowiadać, aby można było łatwo sortować rekordy alfabetycznie. Natomiast wartości liczbowe powinny być umieszczone podobnie jak w arkuszu kalkulacyjnym, czyli sformatowane "do prawej".

Można to zrobić analogicznie jak poprzednio, ale trzeba użyć funkcji RIGHT\$, na przykład tak:

RIGHT\$(" "+STR\$(D), 5)

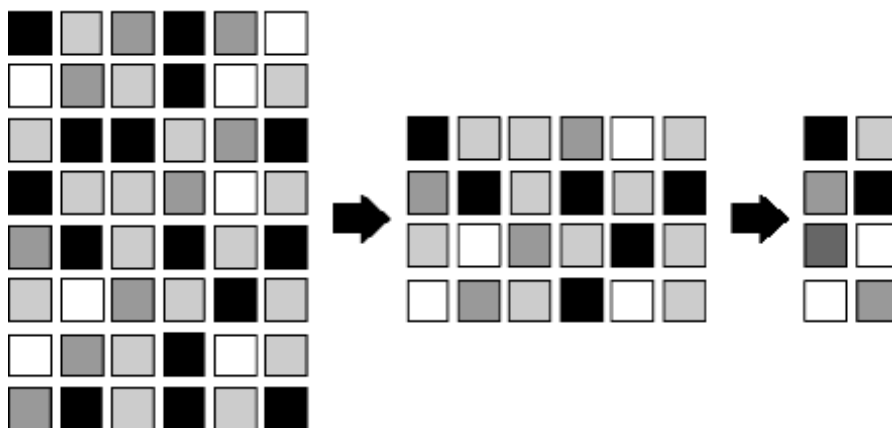
Oczywiście zapis musi być ujęty w ramach jednej zmiennej, a więc na zasadzie "dodawania", na przykład tak:

```
Z$ = LEFT$(A$+" ",
10)+RIGHT$(" "+STR$(D), 5)
```

Wiemy już jak zapisywać dane, ale teraz musimy wiedzieć, jak je odczytywać. Ze zmiennej należy wyodrębnić różne pola. To jest całkiem łatwe, jeśli używamy instrukcji MID\$. Wystarczy podać numer pierwszego znaku i ilość kolejnych, przykładowo tak jak poniżej:

N\$ = MID\$(A\$, 1, 15)

W naszym wypadku ciągi będą uzupełnione o dodatkowe Spacje, dzięki czemu zachowają zawsze określoną długość.



Informacje można zapisać w bardziej skompresowanej formie - jako dane binarne. Oszczędzimy miejsce, ale jednocześnie stracimy możliwość manipulowania za pomocą innych programów. Nasz zapis nie będzie już przypominał danych ASCII.

Podczas zapisywania wartości binarnych musimy zrezygnować ze wszystkich "normalnych" zasad formatowania. Na przykład wartość 13 przechowywana binarnie będzie nie do odróżnienia od znaku RETURN, dlatego nie będziemy mogli użyć instrukcji INPUT, aby ją odczytać.

Jak zapisać informacje w postaci binarnej? Zrobimy to za pomocą instrukcji OPEN. Założmy, że wartości naszych liczb będą wahać się od zera do 65535, a więc zapiszemy ją jako dwa bajty.

Będziemy zapisywać dziesięć liczb do pliku binarnego. Pierwsza linia to "otwarcie" pliku:

```
10 OPEN L, 8, 2, "0:
DATBIN, U, W"
```

Dalej tworzymy pętlę zapisującą odpowiednie wartości:

```
20 FOR Z = L TO 10
30 INPUT D : IF D < 0
OR D > 65536 GOTO 120
40 D% = D/256 : L = D%
* 256
```

Teraz dzielimy liczbę na dwa bajty:

```
50 PRINT #1, CHR$(L);
CHR$(D%);
```

Dalej kończymy pętlę i "zamykamy" plik:

```
60 NEXT Z
70 CLOSE 1
```

Dzięki temu 10 liczb zostało zapisanych jako 20 bajtów. Aby je odczytać korzystamy z instrukcji GET, przykładowo:

```
10 OPEN 1, 8, 2, "DAT-
BIN, U, R"
20 FOR Z = L TO 10
30 GET #1, A$, B$
```

Zawartość zmiennej wyświetlamy w "zwykły" sposób, za pomocą instrukcji PRINT. Oczywiście musimy zakończyć pętlę i zamknąć plik, tak jak poprzednio, czyli dodajemy dwie linie:

```
40 NEXT Z
50 CLOSE 1
```

Liczby zostały zapisane w zwartej formie, dzięki czemu 100 wartości zmieścimy w 1 sektorze na dyskiecie. Jest to bardziej skomplikowane niż zapisywanie plików sekwencyjnych, dlatego polecam używać ten sposób tylko wtedy, gdy jest to konieczne to osiągnięcia odpowiednio małej objętości zapisywanych informacji.

Interfejs RS-232

Od początku swego istnienia mikrokomputer Commodore 64 był bardzo elastyczną platformą w rękach programistów i użytkowników. Dzięki temu zyskał sobie zasłużone miejsce w historii rozwoju informatyki i na pewno w domach wielu z nas. Jak się okazuje, nie stracił zbyt na funkcjonalności pomimo swego dosyć sędziwego wieku. W mniemaniu wielu osób, C64 potrafi dorównać kroku wyżej zaawansowanym platformom, przeobrażając się ze zwykłej zabawki w narzędzie pracy. Z pewnością nie byłoby to wszystko możliwe, gdyby nie porty i interfejsy w jakie zaopatrzyli go projektanci.

To za sprawą tych gniazd, cieszymy się z możliwości słuchania muzyki, drukowania dokumentów, zapisywania danych na nośnikach. Tematem tego artykułu ma być budowa szeroko rozpowszechnionego portu RS-232, który uzupełni listę szerokich możliwości komputera.

Szeregowe przesyłanie bajtów nie jest wynalazkiem szczególnie wydajnym, mimo to jest bardzo często stosowane. Obsługa klawiatury, modemu czy myszy nadal może odbywać się na tym poziomie

i nawet zmiana standardów, w postaci innych odmian portów, nie wpłynęła na zmianę zasady pracy tych urządzeń. Wbrew pozorom duża część urządzeń zewnętrznych od komputera PC będzie także funkcjonowała z Commodore, jeśli wyposażymy się w konwerter poziomów TTL/RS-232. Ta przystawka gwarantuje poprawne przekazanie informacji w obydwie strony i musi być zastosowana, skoro chcemy podłączać osprzęt różny od dedykowanego dla C64

User Port jest zaopatrzony we wszystkie podstawowe linie wymagane przy transmisji szeregowej. Niestety są one w standardzie TTL, czyli poziomy napięcie nie są zgodne z zaleceniami interfejsu RS-232, który pracuje w technologii CMOS.

W technice TTL rozróżniane są napięcia niesymetryczne +5V oraz masa, zaś w CMOS są symetryczne +/-12V oraz masa. Dlatego dla złącza C-64 stosuje się nazewnictwo RS-

232C i w ten sposób dokonuje się podział na dwie kategorie. Zbudowanie interfejsu można zrealizować na kilka sposobów, w tym wypadku poruszymy tylko jeden, a to za sprawą jego kilku niezaprzeczalnych zalet. Poniższy układ jest łatwy do zbudowania (nawet początkujący elektronik nie powinien mieć kłopotów ze złożeniem), wymaga tylko jednego napięcia zasilania +5V (można je pobierać z pinu nr 2 User Portu C-64), zajmuje niewiele miejsca (można go schować w obudowie), nie jest drogi i fantastycznie współpracuje z oprogramowaniem takim jak Novaterm.

Układ MAX 232 wykonany w technologii CMOS zawiera w swej strukturze pojemnościowy podwajacz napięcia z układem inwertera. Z tego właśnie powodu może być zasilany pojedynczym napięciem i spełnia jednocześnie wymagania normy V.28. Jest w stanie pracować z prędkością maksymalną 120 kbit/s i 200 kbit/s dla modelu MAX 232A. Według aplikacji producenta, kondensatory

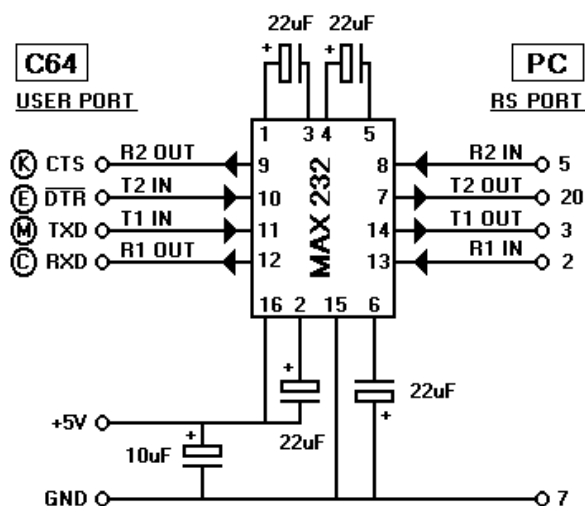
Spis elementów potrzebnych do zbudowania interfejsu:

- układ scalony MAX 232
- kondensator elektrolityczny 10uF 16V
- kondensatory elektrolityczne 22uF 16V (4 szt.)
- gniazdo DB25 (żeńskie)
- wtyk do USER PORT
- obudowa DB25
- obudowa interfejsu (opcjonalnie)
- przewód 4-żyłowy w ekranie (ok. 2m)



zewewnętrzne powinny mieć pojemność przynajmniej 1uF, ale śmiało można ją zwiększyć do około 10uF i nie będzie to w niczym przeszkadzać. Interfejs według założeń normy powinien prawidłowo funkcjonować przy długości przewodu rzędu kilkunastu metrów lub więcej (od strony PC). Ale przy tak długich połączeniach przewód musi być w oplocie, czyli tzw. ekranie, który skutecznie zmniejszy powstawanie przekłamań. Trzeba pamiętać, że zbyt długie połączenia będą przyczyną zmniejszenia prędkości przesyłu, stąd najlepiej ograniczyć je na tyle, na ile jest to potrzebne.

Po stronie TTL, nie możemy już sobie pozwolić na taki luksus i tutaj rygorystycznie zmniejszamy odległość do maksymalnie kilkunastu centymetrów. Dlatego może wygodniej będzie wmontować układ wewnątrz obudowy C-64 i wyprowadzić wtyczkę na tył obudowy. Podłączanie w tym momencie innych urządzeń do User Portu może być trochę ryzykowne, bo będą podpięte równolegle do portu, poza tym wydajność prądowa pinu 2 to tylko 100mA. Spójrzmy teraz na schemat:



Po lewej stronie widzimy linie CTS, /DTR, TXD, RXD. Dwie pierwsze z nich służą do sprzętowego potwierdzenia gotowości odebrania danych (Clear To Send) i gotowości terminalu do transmisji (Data Terminal Ready).

Ogólnie przekazują sygnały sterujące, czyli potwierdzenia typu Handshake. Rzeczywiste dostarczanie danych realizują linie TXD (dane wysyłane) i RXD (dane odbierane). Są nazywane liniami transmisji i odbioru danych.

Nie są to wszystkie dostępne linie interfejsu 232, ale Commodore nie korzysta z pozostałych, więc z powodzeniem można ograniczyć się do wcześniej podanych, a nawet do dwóch: RXD i TXD. Komputery PC doczekały się wielu programów do obsługi transmisji szeregowej. W przypadku C-64 mamy do dyspozycji bardzo solidny program Novaterm, w którym mamy sporo opcji konfiguracyjnych. Najważniejsze, aby oba terminale były ustawione według tej samej konfiguracji. W przeciwnym razie wystąpi konflikt i do połączenia nie dojdzie. Ustawienia programów przedstawiam w ramce powyżej. Taka konfiguracja

Prawidłowe ustawienia programów transmisyjnych:

Dla PC:

- port: COM2
- prędkość: 2400 bodów
- bity danych: 8
- kontrola parzystości: parzystość (lub jej brak)
- bity stopu: 1
- kontrola RTS/CTS :brak
- emulacja terminala: VT 100 (lub ANSI)
- typ terminala: Xmodem-1K (lub Kermit, Punter, Xmodem)
- ustawienia terminala: echo włączone

Dla C-64:

- prędkość: 2400 bodów
- bity danych: 8
- kontrola parzystości: parzystość (lub jej brak)
- bity stopu: 1
- kontrola RTS/CTS: brak
- emulacja terminala: VT 100 (lub ANSI)
- typ terminala: Xmodem-1K (lub Kermit, Punter, Xmodem)
- ustawienia terminala: echo włączone

daje pewne i szybkie połączenie. Działamy z największą dla Commodore prędkością (2400 bitów/s) i transportujemy 8 bitów na raz. Wymagamy, aby następowało sprawdzenie poprawności danych przez kontrolę bitu parzystości, zmniejszamy oczekiwanie bitów stopu do najmniejszej wartości oraz korzystamy z emulacji VT 100, przez co otrzymujemy dokładny obraz tego, co zostało nadane na ekranie drugiego urządzenia. Typ terminala Xmodem-1K spowoduje wysyłanie paczek danych w częściach po 1 kilobajcie. Włączamy echo, aby widzieć na naszym ekranie to, co piszemy na klawiaturze.

Tak pokrótce prezentuje się całe zagadnienie transmisji danych. Jeśli dowolna z wyżej wymienionych opcji nie występuje w użytym programie, konieczne należy odszukać parametry wspólne dla obu programów i sprawdzić poprawność ustawienia. Polecam wykonanie eksperymentów z różnie ustawionymi opcjami. Stare interfejsy mogą okazać się dzisiaj bardziej użyteczne, niż myśli o tym na co dzień.

Taśmy w praktyce

Istnieje wiele metod zapisu danych na taśmie. Najwolniejsza to sposób jakim posługuje się Commodore 64 bez żadnego dodatkowego oprogramowania. Pod względem popularności przodował kiedyś Turbo Tape, który zmniejsza czas ładowania programu. Program 10-minutowy nagrany w turbo zajmuje nieco ponad minutę, a więc jest o co walczyć. Istnieją programy umożliwiające ładowanie programów w systemie turbo. Można je znaleźć na kasetach - są nagrane na samym początku. Warto jednak mieć turbo w kartridżu - ułatwia to pracę, gdyż nie jest wymagane ciągłe ładowanie programu "początkowego".

Komercyjne programy są najczęściej zapisywane we własnym standardzie, wtedy jedyną możliwością zrzućenia programu na dysk peceta jest metoda z plikami TAP. Ładowanie na mikrokomputerze odbywa się poprzez program ładujący, zapisany w zwykły sposób.

Zrzucając taśmę do pliku PRG otrzymamy więc tylko loader, grę zostawiając nierozpoznaną. Wprowa-

dzenie tego do emulatora kończy się wyświetleniem intra lub zostanie wywołany błąd, a następnie zawieszeniem całości pod tytułem "czekam na program".

Pecetowe pliki taśm dzielą się na dwie kategorie: pliki programów, czyli PRG, T64, P00 itp. oraz pliki bezpośredniego zapisu na kasecie, czyli TAP i WAV. TAP jest to zapis danych binarnych dźwięku zoptymalizowany pod kątem taśmy Commodore. Pliki TAP i WAV najlepiej przekonwertować do programów, gdyż mamy z głowy zakłócenia, niestety nie zawsze jest to możliwe. W przypadku komercyjnych gier i programów nie należy tego robić.

Natomiast T64 to plik-kontener zawierający jeden lub kilka plików PRG. Obsługę tych plików ma program StarCommander.

Jeżeli chcemy podłączyć magnetofon Commodore do komputera PC, musimy wykonać interfejs typu C2N-LPT. Przy odrobinie wprawy nie jest to trudne i kosztowne.

Będziemy potrzebować następujące elementy:

- wtyczkę do portu LPT (czyli równoległego), DB25 męską,
- niewielki kawałek uniwersalnej płytki drukowanej, takiej jak do wytrawiania,
- przewody, cynę, lutownicę, kalafonię, pilniki, nóż do tapet lub "nożyki

modelarskie" oraz brzeszczot do metalu,
- źródło 5V DC - zasilacz, komputer, baterie (nie polecane) etc.

Płytkę docinamy pilką tak, aby pasowała do wtyczki Datassette i wystawała o kilka cm na potrzebne przewody i "rękojeść". Na początku docinamy nieco na oko, potem robimy nacięcie klucza, czyli tam, gdzie wchodzi "ząbek" wtyczki.

UWAGA! Wtyczka magnetofonu ma styki po jednej stronie i tam ma być oczywiście miedź, nie laminat. Najlepiej zajrzeć do wtyczki, upewnić się, że wszystko jest w porządku i dopiero wtedy wykonać cięcie. Całość należy oczywiście dopracować pilnikiem.

Teraz wkładamy i wyciągamy kilka razy wtyczkę, dopóki nie zobaczymy na jej krawędzi delikatnych rysek po stykach. Trzeba tak przeciąć miedź na całej długości płytki, aby każda rysa miała swoją ścieżkę za wyjątkiem dwóch rys po dwóch stronach nacięcia klucza. Na początku można ciąć według linijki nożykiem technicznym. Gdy będziemy mieli zarysowany podział, poszerzamy go, aby wyeliminować możliwość zwarcia. Płytkę od strony "użytkownika" trzeba zabezpieczyć kalafonią, bo luty inaczej mogą się nie przykleić.

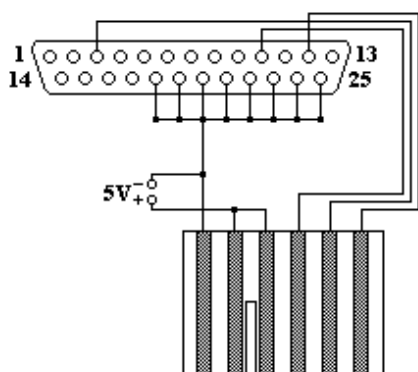
Do szerokiego paska na dwa styki lutujemy jeden przewód i od razu



wyprowadzamy go na zewnątrz - to będzie +5V. Zostanie po jednej stronie jeden samotny styk - tam lutujemy dwa kable - jeden wyprowadzamy jako masę dla 5V, a drugi podłączamy do pinów 18-15 we wtyczce BD25, która powinna mieć małe numery pinów odcisnięte na plastiku. Łączymy je wcześniej kawałkiem drutu.

Patrzac od strony szerokiego styku zostaną trzy przewody. Pierwszy trzeba połączyć z pinem 10, drugi z 3, a trzeci z 12 wtyczki DB25. Kolejność jest liczona od szerokiego styku, czyli ten idący do pinu 12 jest najbardziej z brzegu. Ostatnią operacją jest zlutowanie "na skos" pinów 2 i 15 we wtyczce DB25.

Sprawdzamy, czy nie ma nigdzie zwarc. Kabel może być ekranowany, może nie być. Ja stosuję po prostu cztery druty ze skrętki kat. V, jednak nie polecam tego. Żeby się ustrzec przed błędami wskutek zakłóceń, najlepiej przewody zaekranować.



Są trzy główne metody przenoszenia danych z taśmy:

- przez samplowanie karcie dźwiękowej.
 - przez zgranie surowych danych z Datasette (do pliku TAP).
 - przez zgranie danych do normalnych plików (mam tu na myśli T64 lub PRG).
- Do dwóch ostatnich potrzebny nam sprawny magnetofon oraz adapter C2N-LPT.

Zgrywanie danych z Datasette polega właściwie tylko na podłączeniu magnetofonu przy użyciu wyżej opisanego urządzenia, uruchomieniu odpowiedniego oprogramowania działającego pod czystym DOSem i naciśnięciu w odpowiednim momencie przycisku PLAY.

Do zgrywania poprzez kartę dźwiękową potrzebny nam będzie dowolny magnetofon, najlepiej stereo, może być z korektorem graficznym i przewód pasujący do wyjścia magnetofonu oraz wejścia liniowego karty dźwiękowej.

Ja na swoim Equalizerze ustawiłem następujące parametry:

- 100Hz: -10,
- 1kHz: 0,
- 10kHz +8.

To daje dość dobre efekty, ale można też posłużyć się equalizerem programu do edycji dźwięku. Pierwszy rzut możemy zrobić bez działania equalizera, a potem - jeżeli coś będzie nie w porządku - kolejny zapis wykonujemy z ustawieniami korektora.

Przy użyciu dowolnego programu nagrywamy kasetę do pliku WAV o następujących parametrach:

- 44100Hz,
- 8-bit,
- stereo.

Pamiętajmy też o słupkach miernika przesterowania. Ustawiamy głośność tak, by sygnał był głośny, ale nie powodował zniekształceń. Dalej plik konwertujemy na tryb mono, nie zmieniając niczego innego. Tworzymy więc następujące wariacje:

- Left: 0% , Right: 100%
- Left: 100% , Right: 0%
- Left: 50% , Right: 50%

Po co tyle zachodu? Otóż rzadko zdarza się odczytanie starej kasy od razu mając plik stereo. Program WAV2PRG w ogóle nie zezwala na użycie pliku stereofonicznego, stąd

trzeba wykonać konwersję. Większość plików kodowana jest z użyciem standardu Turbo Tape. Potrzebujemy więc programu TAPE64 dostępnego na poniższej stronie:

http://www.zimmers.net/anonftp/pub/cbm/crossplatform/transfer/datasette/tape64_win32.zip

Działa on z linii komend. Założmy, że pliki WAV są aktualnie w tym samym katalogu, co program TAPE64. Uruchamiamy więc wiersz poleceń, przenosimy się do katalogu po czym wpisujemy:

TAPE64 -i:nazwapliku.wav

Program wyświetli to, co udało mu się znaleźć w pliku WAV. Jeżeli nazwy programów nie są prawidłowe lub program ich zupełnie nie wykrywa, dodajemy opcję:

TAPE64 -i:nazwapliku.wav -r

Jeżeli jesteśmy usatysfakcjonowani wynikami, trzeba je zapisać. Polecenie:

TAPE64 -i:nazwapliku.wav -r -ot:tasma

zapisze tylko działające programy do pliku o nazwie "tasma.t64". To samo trzeba zrobić z pozostałymi plikami, uważając na uszkodzone - nie każdy plik będzie zapisany poprawnie i trzeba o tym pamiętać.

Wyniki można potem złożyć za pomocą programu Star Commander dostępnego pod adresem:

<http://sta.c64.org>

Jeżeli nie możemy zapisać programu polecam dodawać opcję:

-s:0.9

lub

-s:1.1

Powoduje to delikatną manipulację prędkością, która na starszych magnetofonach jest konieczna.

Ponadto wato spróbować jeszcze raz nagrać plik z kasety - na starych kasetach często tworzą się błędy tymczasowe. Przy okazji można ponownie przewinąć kasetę w magnetofonie - to naprawdę działa. Możemy też spróbować zmieniać parametry programowym equalizerem - wyciszać niskie częstotliwości koncentrując się na ok. 6 kHz.

Plik T64 można uporządkować bardzo prosto - wystarczy użyć programu Star Commander, który otwiera te pliki jak katalogi i umożliwia rozpakowanie danych, dodawanie i kopiowanie między plikami T64 poszczególnych programów. Najprościej skopiować T64 do nowego pliku i skasować z kopii całą zawartość. Możemy jej wtedy używać jako tymczasowej przechowalni dla sortowanych programów.

Oprócz TAPE64 istnieje jeszcze program WAV2PRG dostępny na stronie:

wav-prg.sourceforge.net

Jego wtyczki umożliwiają odczyt wielu standardów. Niestety w tym wypadku skuteczność jest niższa niż TAPE64, w dodatku pliki uszkodzone czasem ładują na dysku obok dobrych, a jedyne, co je odróżnia to wpis w logu.

Zajmiemy się teraz czystym trybem MS-DOS, czyli jak to wszystko zrobić przez interfejs C2N na komputerze PC ze starym systemem operacyjnym. Służą do tego trzy programy:

- CBMTT (czyli C64load i C64save) - odczytuje normalny zapis - możemy przyjąć, że nie obsługuje turbo (jakość odczytu turbo jest wątpliwa, a zapis następuje tylko z loaderem):

<http://www.zimmers.net/anonftp/pub/cbm/crossplatform/transfer/datasette/cbmmtt101.zip>

- TTAPE010 (TTLoad oraz TTSave) - działa odwrotnie - obsługuje turbo:

<http://www.zimmers.net/anonftp/pub/cbm/crossplatform/transfer/datasette/ttape010.zip>

- MTap i PTap - tworzą surowe zrzuty w formacie TAP, w związku z czym będziemy mieli tutaj najwięcej pracy:

<http://markus.brenner.de/>

Na początek proponuję wyczyścić głowicę w magnetofonie i sprawdzić skos w C64. Program C64LOAD możemy wywołać z opcją "-v". Po naciśnięciu PLAY pojawi się graficzny podgląd ścieżki, dzięki czemu można korygować skos głowicy. Wykonujemy regulację śrubokrętem do czasu, aż przebiegi będą niewęższe.

Pełna składnia C64LOAD wygląda tak:

C64LOAD plik.prg -a -i1

Parametr "-a" należy zostawić, co spowoduje, że program sam dobierze prędkość. Natomiast "-i1" oznacza, że magnetofon jest podłączony do portu LPT1.

Program poprosi o naciśnięcie PLAY, a następnie pojawi się raport z kopiowania programu.

TTLOAD obsługuje się podobnie, ale jest prostszy w obsłudze. Po wykonaniu instrukcji:

TTLOAD plik.prg

program poprosi o PLAY i skopiuje plik. Komunikat "Ready" znaczy, że operacja przebiegła poprawnie. "Load error" to oczywiście błąd. Podczas operacji na magnetofonie wszystko jest blokowane (z klawiaturą włącznie), a jedyne co przerywa działanie programu, to naciśnięcie STOP na magnetofonie.

Instrukcję TTLOAD możemy umieszczać po kolei w pliku wsadowym, na przykład tak:

```
echo 01 >>raport.txt
TTLOAD 01.prg >>raport.txt
echo 02 >>raport.txt
TTLOAD 02.prg >>raport.txt
echo 03 >>raport.txt
TTLOAD 03.prg >>raport.txt
echo 04 >>raport.txt
TTLOAD 04.prg >>raport.txt
```

co spowoduje, że po uruchomieniu i naciśnięciu PLAY, program sam skopiuje kolejne programy, a raporty wraz z nazwami programów powędrują do pliku "raport.txt". Zatrzymanie skryptu odbywa się przez STOP, CONTROL-C, a później - PLAY-STOP na magnetofonie.

Za pomocą programu MTAP można tworzyć surowe obrazy w taki sposób:

MTAP plik.tap

lub z większym buforem pamięci:

MTAP -buffer 16 plik.tap

Poi wykonaniu kilku takich samych pod względem długości i sumy kontrolnej CRC, możemy skończyć i załadować je do programu FinalTAP:

<http://www.coder.pwp.blueyonder.co.uk/>

Za pomocą opcji "Optimize" można zmniejszyć objętość i ilość zakłóceń. Gotowe pliki najlepiej sprawdzać na emulatorze.

Przed zapisaniem danych na taśmę proponuję najpierw ją skasować. Wbrew pozorom nie jest to łatwe, bowiem większość magnetofonów przenosi na kasowaną taśmę drgania zasilania lub szum, co negatywnie odbija się na całości nagrania.

Osobiście próbowałem kilka magnetofonów, w tym: Datasette, MK-225, MK-450, Daewoo, Turbocorder oraz Unitra M3018. W rezultacie tylko ten ostatni odpowiednio rozmagnesował nagranie. W pozostałych przypadkach pozostawał mniej lub bardziej słyszalny szum, buczenie lub ciche poprzednie nagranie. Jeżeli



mamy pewny magnetofon, możemy oczywiście zdecydować się za zapis przy jego użyciu. Mamy następujące możliwości:

- zapisywać poprzez plik audio w formacie WAV, kartę dźwiękową i zwykły magnetofon.

- zapisywać poprzez plik w formacie TAP i interfejs C2N (polecam do nagrania turbo),

- zapisywać poprzez plik w formacie PRG i interfejs C2N.

Zapis pliku WAV wygląda następująco:

1. Uruchamiamy WAV2PRG.

2. Wybieramy drugą pozycję "Convert PRG, P00 or T64 to TAP or WAV".

3. Wybieramy rodzaj zapisywanego programu (kernal, turbo, turbo z loaderem), typ mikrokomputera i ustawiamy cel jako WAV.

4. Wybieramy przycisk OK, a następnie plik do zapisu. Jeżeli nie jest to T64, z którego można wybrać pozycję z ramki (niżej), należy wpisać nową nazwę w dolnej części okna. Inaczej można się pogubić, bo każdy program będzie wyświetlony bez nazwy.

5. Zatwierdzamy wybór i zapisujemy w zwykły sposób plik w formacie WAV.

Zapis pliku TAP wygląda podobnie, trzeba tylko wskazać jako cel inny format

Nagrywanie plików WAV nie jest trudne. Łączymy wejście magnetofonu z wyjściem karty dźwiękowej i włączamy odtwarzanie pliku WAV. Praktyka wygląda tak, że trzeba jeszcze eksperymentalnie dobrać poziom głośności. Osobiście polecam monofoniczne magnetofony MK-450 oraz MK-22x. Należy przyłączyć dźwięk przez wtyczkę DIN i uważać na przewód zasilający, który może

powodować zakłócenia. Nagrywanie plików TAP w systemie MS-DOS wygląda tak:

```
ptap plik.tap
```

lub

```
ptap -lpt1 plik.tap
```

Następnie naciskamy RECORD-/PLAY na magnetofonie. Sposób ten można stosować zarówno do oryginałów w TAP, jak i do samodzielnie wykonanych plików.

Aby nagrywać plik PRG trzeba użyć innej formuły:

```
C64SAVE PLIK.PR
```

Dalsze instrukcje są dość zrozumiałe. Nie wprowadzamy zmian, numerami ustalamy sposób zapisu ("Basic File") i potwierdzamy chęć rozpoczęcia transferu. Gdy program poprosi, wciskamy RECORD/PLAY.

Zapis jako TURBO TAPE daje nam zapis wraz z loaderem, co zajmuje trochę czasu i taśmy.

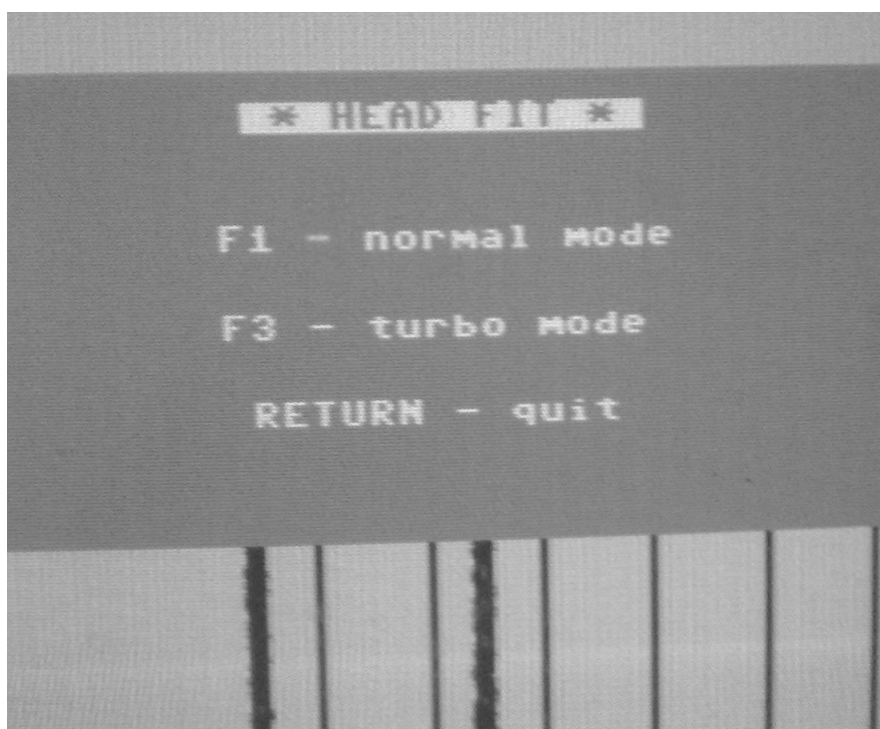
Nagranie można sprawdzić za pomocą programu C64LOAD lub TT-

LOAD. Ten drugi używamy w sytuacji, gdy zapisujemy dane w systemie turbo. Zapisujemy tymczasowy plik i obserwujemy czy nie ma błędów.

Programy w turbo z loaderem, utworzone programem WAV2PRG, musimy sprawdzić dwukrotnie. Pierwszy raz przy użyciu C64LOAD, gdzie otrzymamy krótki program z wpisaną przez nas nazwą. Drugi raz - za pomocą TTLOAD, gdzie otrzymamy właściwy program.

Możemy też spotkać kasety magnetofonowe z głowicą w środku, służące do słuchania muzyki z odtwarzaczy samochodowych MP3. Sprzęt taki może być umieszczony w magnetofonie i umożliwia wtedy całkiem pewny przesył od komputera PC do C64.

Jedyne, o czym należy pamiętać, to wyprowadzenie przewodu i odpowiednie ustawienie głośności w komputerze. Programy przesyłamy przez WAV2PRG tak, jak tworzyliśmy pliki WAV, lecz jako cel wybierając "Sound". Uwaga na programy TAP - może być wymagana pauza, przykładowo na dekompresję programu, której nie będzie można wywołać.



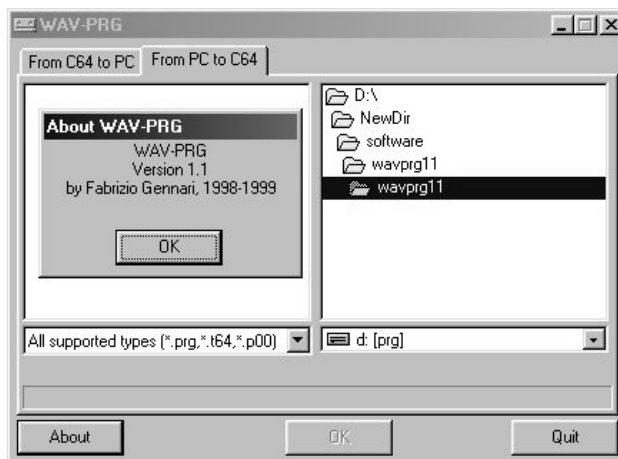


Programy jednoczęściowe można za to przesyłać bezpiecznie. Aby przegrać dane ze stacji 1541 na magnetofon, najlepiej zrobić to przy użyciu Star Commandera. Z pliku D64 kopiujemy plik PRG i nagrywamy go na taśmę, sprawdzając wcześniej (na przykład na emulatorze), czy działa. Ten sposób nie zawsze będzie skuteczny, dlatego można zrobić zapis zrzutu pamięci, czyli stanu po uruchomieniu gry. Można to zrobić dość łatwo za pomocą emulatora WinVICE lub podobnego. Będziemy potrzebować:

- emulator obsługujący cartridge Action Replay,
- obraz ROMu cartridge'a Action Replay (polecam 6.0).
- program WAV2PRG w najnowszej wersji.
- dodatek dla WAV2PRG umożliwiający odczytanie programów zapamiętanych przez Action Replay. Kopiujemy go do katalogu o nazwie PLUGINS.

Dalej wykonujemy następujące czynności:

1. Uruchamiamy emulator skonfigurowany z Action Replay.



2. W emulatorze tworzymy pusty plik TAP i montujemy go jako Datassette.

3. Montujemy obraz dyskietki z grą, ładujemy grę i uruchamiamy tak, by znaleźć się w położeniu startowym (na przykład w pierwszym poziomie).

4. Używamy opcji "Cartridge Freeze" i przy użyciu menu Action Replay zapisujemy na kasety zrzut.

5. Mając gotowy zrzut wyłączamy emulator.

6. Plik TAP otwieramy przy pomocy WAV2PRG, przy starcie wybieramy pierwszą opcję, a dalej plugin Action Replay. Po wykonaniu analizy pliku TAP, program znajdzie nasz zapis i

utworzy jego kopię w "zwykłym" formacie PRG.

7. Testujemy zapis w emulatorze do momentu, w którym się zawiesi lub zacznie niepoprawnie wyświetlać grafikę albo mapę. Oznacza to, że trzeba odczytać kolejne dane z dyskietki. Od tego momentu możemy zrobić kolejny zrzut, aby kontynuować rozgrywkę.

Warto dodać, że co jakiś czas powinniśmy wykonać czynności konserwacyjne. Kasyety najlepiej regularnie przewijać. Zapobiega to rozmagnesowywaniu taśmy.

Warto również sprawdzać rolki. W tym celu włączamy magnetofon, otwieramy kłapkę i wciskamy PLAY. Sprawdzamy, czy rolka się obraca. Jeżeli nie, a pomiędzy nią, a srebrną ośką jest przerwa - znaczy to, że rolkę należy wymienić. Jes to najczęstsza usterka w Turbocorderach, bowiem słaba rolka pod wpływem czasu i wilgoci po prostu parcieje. Jeżeli tak jest - trzeba

spróbować zatrzymać ją na chwilę palcem. Jeżeli zatrzymuje się łatwo - rolka nadaje się do czyszczenia, zmatowienia lub wymiany.

Jeżeli chcemy wykorzystać stare kasyety, a przeszkadzają nam stare napisy, nie należy używać nożyka, ani acetonu czy benzyny. Najlepiej zastosować bardzo delikatny papier ścierny.

Mam nadzieję, że moje uwagi będą dla Was przydatne. Kasyety nie są dzisiaj często używane, ale prawdziwego fana retro dźwięk magnetofonu będzie z pewnością miał najlepszy klimat. Tym bardziej warto przypomnieć sobie, jak trzeba się z nim obchodzić.



Power at Sea

W 1988 roku firma Accolade wydała ciekawą, acz dzisiaj mało znaną grę symulacyjną. Większość graczy zna hity takie jak "Test Drive" czy "Elvira", natomiast tytuły "historyczne" zwykle pozostają bardziej z boku. W tym numerze naszego magazynu piszemy także o grach dla 8-bitowych modeli komputerów marki Apple, a ten artykuł jest swoistym uzupełnieniem tekstu ze strony numer 65.

Fabula "Power at Sea" rozgrywa się w czasie II wojny światowej, podczas bitwy w Zatoce Leyte, która miała miejsce w dniach 23–26 października 1944 roku na Filipinach. Gracz zostaje dowódcą amerykańskiej grupy okrętów składających się z lotniskowca, pancernika typu i statku transportowego. Celem gry jest zabezpieczenie zatoki Leyte w ciągu 96 godzin, co jest rozumiane jako zdobycie czterech japońskich baz. W grze zdobywa się punkty za zdobyte bazy, zatopione okręty, zestrzelone samoloty oraz zachowane zasoby.

Jako dowódca amerykańskiej floty, zarządzamy nią za pomocą decyzji wydawanych z mostka kapitańskiego. Do dyspozycji mamy telegrafistę, nawigatora, osobę monitorującą uszkodzenia jednostek, a także kolejną osobę odpowiedzialną za działania wojenne. Przed rozpoczęciem rozgrywki gracz decyduje o zasobach, jakie zamierza wykorzystać

w akcji, czyli o liczbie bombowców, myśliwców, żołnierzy oraz ilości paliwa. Po rozpoczęciu gry, w miarę poruszania się floty po mapie, napotykamy na swojej drodze nieprzyjacielskie samoloty, a także okręty oraz zdobywamy wrogie bazy.

W czasie kontaktu z wrogiem gra zmienia się z symulacyjnej w grę zręcznościową z perspektywy pierwszej osoby, w której możemy sterować bombowcami i myśliwcami z pozycji pilota, artylerią główną pancernika, działem przeciwlotniczym, a także artylerią główną pancernika. Kluczowym elementem decydującym o końcowym sukcesie jest jednak zdobywanie baz wroga. Odbywa się to poprzez próby zniszczenia wrogich stanowisk przy pomocy artylerii oraz wysłanie odpowiedniej liczby żoł-

nierzy na desant. Bazy mają trzy wyjściowe poziomy trudności, które zmniejszają się wraz ze skutecznym ostrzałem.

W mojej ocenie gra jest bardzo udana, niestety została przyjęta niezbyt dobrze. Krytykowano głównie aspekt strategiczny rozgrywki oraz zbytne skupienie się na części zręcznościowej, co spowodowało niedopracowane elementów symulacyjnych. "Power at Sea" ma za to bardzo dobrą grafikę oraz ciekawe efekty dźwiękowe.

Być może nie pogramy w nią zbyt długo, ale każde uruchomienie osobiście przynosi mi bardzo wiele przyjemności. A przecież o to chodzi w naszym hobby.

Opracował: Łukasz Nowak



SuperPET i większa pamięć

Komputer SuperPET nie jest popularnym sprzętem retro i tym bardziej postanowiłem pokazać jego nietypowe możliwości.

SuperPET posiada dodatkowe 64 kilobajty pamięci. Ma 16 banków po 4 kilobajty, które znajdują się od adresem #9000 (wartość szesnastkowa). Przy zastosowaniu odpowiedniej linii z instrukcją POKE, możemy spowodować włączenie odpowiedniego banku pamięci. Trzeba mianowicie użyć adresu 61436 i wartości od 0 do 15.

Pełne wykorzystanie pamięci jest możliwe, gdy SuperPET jest używany w trybie Commodore BASIC. Dzięki posiadaniu głównego programu w pamięci PET (32 kilobajty) można uzyskać dostęp do 16 modułów po 4 kilobajty lub procedur BASIC, które są wstępnie ładowane do dodatkowych banków pamięci. Aby to zrobić, w programie musimy wykonać trzy główne elementy. Pierwszym jest użycie instrukcji POKE z adresem 61436, drugim - wykonanie:

POKE 41,144

co spowoduje, że program zostanie ustawiony na adres #9000. Na koniec trzeba użyć instrukcji GOTO, aby uruchomić konkretną linię programu.

Zauważmy, że zmienne programu pozostaną w zwykłym obszarze 32 kilobajtów pamięci, a więc będą mogły być używane w uniwersalny sposób. Zmienne tekstowe powinny być tworzone przez główny program tak, aby dane nie znalazły się pod adresem wyższym niż #8000. Wszystkie zmienne łańcuchowe powinny zostać zainicjowane przez główny program. Zapewni to, że żadne ciągi wskaźników nie wskażą więcej niż 8000 dolarów. Aby powrócić z modułu do głównego programu, trzeba wpisać:

POKE 41,4

a potem GOTO do wskazania linii, tym razem w programie głównym.

Moduły są zapisywane i ładowane do normalnej pamięci PET, czyli wspomnianego obszaru 32 kilobajtów. Aby przenieść moduł do banku 4-kilobajtowego, można użyć krótkiego kodu w Asemblerze. Musi być on wczytany do pamięci na początku, za pomocą monitora. Po wykonaniu określonego modułu, za pomocą instrukcji POKE 61436 ustawiamy numer banku. Dalej wykonujemy SYS 32688. Analogicznie możemy załadować wszystkie moduły i zresetować system poprzez SYS64790. Wczytany i uruchomiony zostanie główny program.

```
LDA #$00
STA $00
LDA #$04
STA $01
LDA #$00
STA $02
LDA #$90
STA $03
LDX #$0F
LDY #$00
LDA ($00),Y
STA ($02),Y
DEY
BNE $7FC4
INC $01
INC $03
DEX
BMI $7FD5
JMP $7FC2
LDA #$90
STA $29
JSR $B4B6
LDA #$04
STA $29
RTS
```

Na koniec mała uwaga - próba sprawdzenia, który bank jest włączony poprzez PEEK 61436 w tej sytuacji nie jest możliwy. W ramce przedstawiam listing programu w Asemblerze, który przenosi program modułu z głównej pamięci do adresu #9000 i sprawia, że można go uruchomić.



Przenoszenie danych na C64

Jeżeli zastanawiacie się, w jaki sposób wykonać konwersję dyskietek do plików D64, przeczytajcie ten artykuł. Gotowe pliki można pobrać z Internetu, jeśli jednak mamy własne archiwum, nie warto go tracić. Gotowe pliki można wykorzystać przy pomocy interfejsów typu SD2IEC oraz na emulatorach.

Plik w formacie D64 to skomplikowana kopia całej zawartości dyskietki 5,25" ze stacji typu 1541, zapisanej w pojedynczym archiwum. Chciałbym tu zwrócić uwagę, że chociaż nośniki magnetyczne są uważane za ograniczone jeśli chodzi o czas przechowywania, to osobiście nie miałem nigdy problemów z odczytywaniem starych dyskietek, nagranych jeszcze w latach '80-tych. Z pewnością ich czas żywotności kiedyś się skończy, tym bardziej nie polecam czekania.

Istnieje kilka metod, które można wykorzystać, aby stworzyć obrazy dyskietek z oryginalnych nośników. Pierwsza polega na połączeniu komputera PC z odpowiednim przewodem do portu równoległego (ang. parallel, czyli popularne LPT), który może współpracować

z napędem 1541. Następnie instalujemy oprogramowanie, które umożliwia odczytywanie dyskietek na stacji Commodore, dzięki czemu pliki dla C64 można kopiować bezpośrednio na dysk komputera z systemem Windows lub Linux.

Można tu wykorzystać przewody typu XM1541 i oprogramowanie Star Commander, na przykład dostępne na tej stronie:

<http://sta.c64.org/sc.html>

Niestety sposób ten jest dość awaryjny, przynajmniej z mojego doświadczenia. Transmisja danych potrafi być przerywana, ponadto nie każda dyskietka jest poprawnie odczytywana, tak jak na prawdziwym C64.

Kolejna metoda przenoszenia dyskietek polega na użyciu modemu Com-

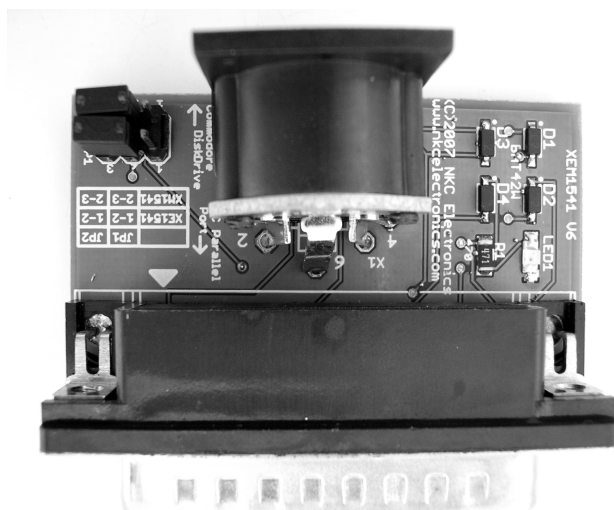
modore i kabla typu RJ-45, tak zwanego "krosowanego" (crossover). Dzięki temu można połączyć się jak za starych, dobrych czasów - za pomocą dwóch modemów i programu terminalowego. Dodajmy, że obecnie można kupić małe modemy USB, na przykład znanej niegdyś firmy USRobotics, a obsługa urządzeń sięga także nowych systemów jak Windows 10. Więcej szczegółów znajdziemy tutaj:

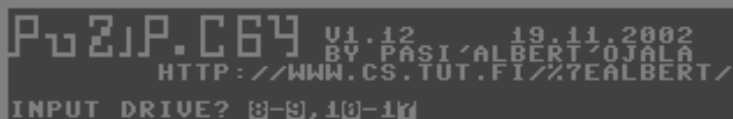
<http://www.usr.com/products/56k-dialup-modem/usr5637/>

Po stronie C64 możemy użyć programu Bobstern Pro, który jest prosty, ale pozwala na transmisję plików. Jest to jednak bardzo żmudne i czasochłonne. Odpowiedniego modelu telefonicznego nie znajdziemy łatwo, ale warto wiedzieć o takiej możliwości. Może wśród starych "rupieci" znajdą się urządzenia, które reanimujemy i z powodzeniem wykorzystamy przy naszym Commodore.

Następny sposób na przeniesienie danych to zakup karty adaptera Ethernet dla C64. Można go zobaczyć na wielu stronach w sieci, na przykład tutaj:

<http://www.go4retro.com/products/64net-ic/>





```
PuZIP.C64 V1.12 19.11.2002
BY PASI 'ALBERT' OJALA
HTTP://WWW.CS.TUT.FI/%7EALBERT/
INPUT DRIVE? [0-9], [0-14]
```

pusty ciąg tekstowy. Oczywiście potwierdzamy wszystko klawiszem RETURN.

Podczas odczytywania lub zapisywania danych ekran zostanie wygaszony, co przyspiesza kompresję. Na ekranie będą widoczne wielokolorowe pasy. Cała operacja może potrwać ok. 15 minut na standardowej dyskietce w stacji 1541, a więc warto zrobić sobie małą przerwę na kawę.

Warto dodać, że program "PuZIP" działa również w wersjach dla C128, Plus/4, a nawet VIC-20. Jest uniwersalnym narzędziem, które potrafi skorzystać z 80-kolumnowego trybu ekranu i rozszerzeń typu REU czy

Na tej samej stronie znajdziemy oprogramowanie, które pozwoli połączyć się z siecią. Co prawda nie ma zbyt dużych możliwości, ale można połączyć się z siecią, na przykład na pomocą Geosa, a nawet wejść na kanał IRC. Uruchomimy też program terminalowy, który do transmisji plików będzie doskonały (opcje "Upload" i "Download"). Przy okazji polecam zobaczyć tekstową przeglądarkę stron WWW.

Ostatnią i najwygodniejszą metodą jest zakup interfejsu typu SD2IEC i uruchomienie programu pozwalającego wykonać kompresję dyskietek do pliku D64. Nosi on nazwę "PuZIP" i znajdziemy między innymi na następującej stronie:

<http://csdb.dk/release/?id=44012>

lub

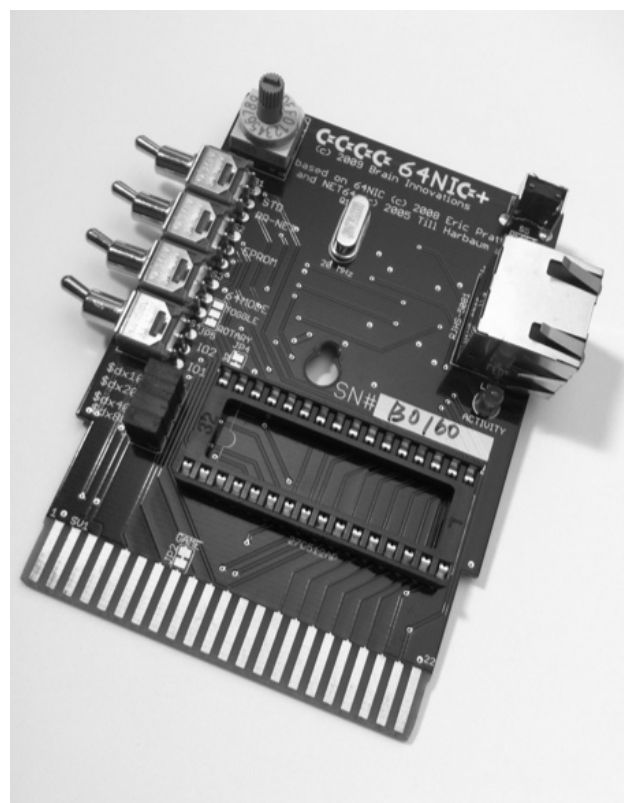
<http://commodore.software/downloads/category/114-puzip>

Po uruchomieniu program najpierw pyta o napęd wejściowy (czyli źródłowy). Należy nacisnąć odpowiednio 8 lub 9, aby wybrać napęd. Dalej wybieramy pliki, które mają być

archiwizowane, naciskając klawisz Y dla kompresji LZ77 lub S, aby zapisać dane bez kompresji. Można także użyć 1-3, aby skorzystać z szybszej, lecz mniej wydajnej kompresji.

Następnie naciskamy przycisk D, aby uruchomić kompresję obrazu dyskietki. Typ obrazu jest automatycznie wykrywany i wybierany między D64, D71 oraz D81. Program pozwala filtrować pliki, które znajdują się w docelowym (wyjściowym) obrazie D64. Jeśli nie chcemy, aby plik był w archiwum, naciskamy N.

Po wybraniu wszystkich plików można nacisnąć klawisz Q, aby pominąć pozostałe. Dalej wybieramy napęd wyjściowy. Wyświetlona zostanie nazwa archiwum, można też podać komentarz. Jeśli nie chcemy żadnego komentarza, pozostawiamy



IDE64 czy SuperCPU. Używa też trybu "burst" w Commodore 128. Jednocześnie podstawowa wersja działa na zwykłych, nierozbudowanych modelach.

Opracował: Łukasz Nowak



Śledzenie działania

Śledzenie wykonywania programu jest bardzo przydatne jako narzędzie do wykrywania różnych błędów, czyli do debugowania. Pozwala to sprawdzić, na którym numerze linii zatrzymał się program. Programista może zobaczyć błędy występujące na bieżąco, a także dużo łatwiej znaleźć obszary, w których występują potencjalne problemy.

Program widoczny obok dostosowuje się do dowolnej wielkości pamięci w VIC-20, a sam zużywa tylko ok. 200 bajtów pamięci. Po zapisaniu i uruchomieniu, w ciągu kilku sekund na ekranie powinien pojawić się komunikat w formie wartości SYS, które pozwolą włączyć i wyłączyć śledzenie. Na nierozbudowanym VIC zobaczymy wartości jak poniżej:

```
TRACE ON SYS(7501)
TRACE OFF SYS(7488)
```

Jeżeli po wykonaniu SYS(7501) komputer zawiesi się lub będzie wykonywał nieoczekiwaną operację oznacza

to, że program został błędnie wpisany. Pamiętajmy, że usuwanie usterek z programu może polegać nie tylko na rozpoznawaniu błędów składniowych, ale także logicznych. Niezgodność działania może polegać na przykład na przypisaniu błędnego symbolu zmiennej, wartości początkowej lub wywołanie algorytmu z nieprawidłowymi parametrami. Schemat wskazania błędu jest podobny na każdym komputerze, różnią się tylko dostępne narzędzia pomocnicze. Dlatego warto nauczyć się to robić na najprostszych maszynach, aby pewnie rozpocząć pracę na lepszych konfiguracjach.

```
5 F = 0 : C = PEEK (55) -192 : IF C<0 THEN C = C + 256 : F = -1
10 D = PEEK (56) + F : POKE 55, C : POKE 56, D : CLR
15 N = PEEK (55) + 256 * PEEK (56)
20 F = 0 : FOR D = N TO N + 191 : READ A$ : IF ASC (A$)<58 THEN A = VAL (A$) :
GOTO 35
25 IF ASC (A$) = 76 THEN A = VAL (RIGHT$ (A$, LEN(A$) -1)) + PEEK (55) : IF A>255
THEN A = A - 256 : F = 1
30 IF ASC(A$) = 72 THEN A = VAL (RIGHT$ (A$, LEN(A$) -1)) + PEEK (56) + F : F = 0
35 POKED, A : NEXT
40 PRINT "TRACE ON SYS("N + 13")"
45 PRINT "TRACE OFF SYS("N")" : NEW
50 DATA 169, 230, 133, 115, 169, 122, 133, 116, 169, 208, 133, 117, 96, 169, 255,
141, 61, 3, 169, 76
55 DATA 133, 115, 169, 131, 133, 116, 169, 10, 133, 117, 96, 72, 138, 72, 152, 72,
165, 58, 201, 250
60 DATA 176, 12, 205, 61, 3, 208, 10, 165, 57, 205, 60, 3, 208, 3, 76, 1134, 10,
165, 57, 141
65 DATA 60, 3, 141, 62, 3, 165, 58, 141, 61, 3, 141, 63, 3, 169, 18, 32, 210, 255,
169, 32
70 DATA 32, 210, 255, 169, 0, 141, 64, 3, 162, 0, 32, 1148, 10, 173, 65, 3, 240,
3, 238, 64
75 DATA 3, 173, 64, 3, 240, 8, 173, 65, 3, 9, 48, 32, 210, 255, 232, 224, 5, 208,
227, 173
80 DATA 64, 3, 208, 5, 169, 48, 32, 210, 255, 169, 146, 32, 210, 255, 104, 168,
104, 170, 104, 230
85 DATA 122, 208, 2, 230, 123, 76, 121, 0, 169, 0, 141, 65, 3, 56, 173, 62, 3,
253, 1182, 10
90 DATA 168, 173, 63, 3, 253, 1187, 10, 144, 12, 238, 65, 3, 141, 63, 3, 140, 62,
3, 76, 1153
95 DATA 10, 96, 16, 232, 100, 10, 1, 39, 3, 0, 0, 0, 32, 56, 53, 32, 4, 1, 20, 1
```

Łączenie programów na VIC-20

Dodawanie do siebie dwóch programów w celu utworzenia jednego, spójnego listingu jest całkiem proste. Jedną w pozytywnych cech Basic'a jest automatyczne relokowanie programów podczas wczytywania. Program jest automatycznie wprowadzony ze właściwe miejsce w pamięci RAM. Dodajmy, że mogą to być trzy różne obszary pamięci, w zależności od posiadanej ilości w komputerze.

BASIC na sprężenie z wolną pamięcią 3,5 KB domyślny adres początkowy to 4097. Wszystkie programy są zapisywane z tym adresem jako punkt wyjścia. Na komputerze rozbudowanym o dodatkowe 3 KB początkowy adres wynosi 1025. Punkt początkowy w Basic'u może więc się zmieniać. Jak to działa? Najpierw sprawdzane jest, gdzie zaczyna się Basic. Jest to adres, gdzie relokator rozpocznie przechowywanie dowolnego programu, który zostanie wczytany.

Relokator działa różnie w zależności od wskaźnika lokalizacji obszaru "początku Basic'a", dlatego możliwe jest połączenie dwóch oddzielnych programów. Należy jednak pamiętać, że nie mogą one posiadać tych samych numerów linii, a program znajdujący się w pamięci musi mieć nu-

mery linii niższe niż program, który będziemy dołączać jako drugi. Gdy już mamy dwa programy - jeden zapisany, a drugi w pamięci komputera możemy zabrać się do pracy. Czyścimy ekran i wpisujemy następującą linię w trybie bezpośrednim:

```
PRINT PEEK(43),PEEK(44)
```

Na komputerze wyposażonym w 3,5 KB RAM otrzymamy wartości - odpowiednio 1 i 16. Trzeba je zapisać lub zapamiętać, bo będą potrzebne później. Teraz wpisujemy kolejne instrukcje:

```
POKE 43, PEEK(45)-2
```

```
POKE 44, PEEK(46)
```

Powyższe linie ustawiają wskaźnik początku programu Basic'a na koniec bieżącego listingu, czyli tego, który znajduje się aktualnie w pamięci. Następnie wczytujemy drugą część programu, jaka ma zostać dołączona - w zwykły sposób, czyli za pomocą polecenia LOAD. Zanim oba pro-

gramy zostaną połączone trzeba jeszcze zresetować "początek Basic'a" do adresu, który był aktywny przed wczytaniem drugiego listingu. Aby to zrobić wykonujemy dwie linie POKE wpisując wartości, które wcześniej zapisaliśmy. Na komputerze z 3,5 KB pamięci wygląda to tak:

```
POKE 43,1
```

```
POKE 44,16
```

Jeśli wszystko zostanie wykonane poprawnie, cały program powinien być już możliwy do wyświetlenia. Łączenie programów może być bardzo przydatne, bo pozwala opracować kilka niezależnych zestawów podprogramów, a następnie dołączyć go do głównego listingu. Dzięki temu możemy na przykład utworzyć własną bibliotekę często używanych podprogramów, użyć wysokich numerów linii, a potem łączyć poszczególne części, aby znalazły się w głównym programie. Oszczędzamy wtedy czas na wpisywanie podobnych fragmentów kodu i pracujemy dużo szybciej.





Optymalizacja programów w Atari Basic

W tym artykule chcę za- stanowić się nad sposobem pisania programów eleganckich i efektywnych. Omawiane tutaj zagadnienia nabierają szczególnego znaczenia przy pisaniu dużych programów, ale poniższe uwagi warto jest stosować na co dzień, aby stały się dobrym nawykiem. Rozważania nasze dotyczyć będą w zasadzie języka Atari Basic, ale ich charakter jest ogólny i można je wykorzystać programując w innym języku.

Jeżeli pisany przez nas program jest duży i skomplikowany to w czasie jego uruchamiania będzie on wielokrotnie czytany i może wymagać wprowadzenia wielu zmian. Często do pracy nad programem powracamy po dużej przerwie lub też program pisany przez nas jest uruchamiany bądź modyfikowany przez innego użytkownika.

Dlatego należy założyć, że odbiorcą programu jest przede wszystkim człowiek, a w następnej kolejności komputer. Przejrzystości programu należy podporządkować inne mniej ważne kryteria, jak na przykład jego sprawność rozumianą jako szybkość wykonywania programu. Stosowanie podanych poniżej wskazówek znacznie ułatwi nam pis-

anie przejrzystych programów, które będzie później łatwo uruchamiać i poprawiać.

Starajmy się stosować znaczące nazwy zmiennych. Nie używajmy nazw takich jak WWWX13 czy AAAA. Używanie nazw opisowych zwiększa czytelność programu. Cyfry umieszczajmy na końcu nazwy. Pamiętajmy przy tym, że cyfry 1, 0, 2, 5 mogą mylić się z literami I, O, Z, S. Nie używajmy słów kluczowych języka jako nazw zmiennych. Poniżej przedstawiam przykład. Mimo, że linie te są w Basicu poprawne, są bardzo nieczytelne i mogą wprowadzać w błąd:

```
LET IF=1
LET THEN=75
LET GOTO
IF IF=THEN THEN GOTO
GOTO
```

W miarę możliwości unikajmy stosowania zmiennych roboczych. Mimo, że są one często potrzebne, nigdy nic twórzmy ich zbyt wiele. Duża liczba zmiennych roboczych zwiększa ilość szczegółów, które trzeba analizować przy czytaniu programu.

Nic zmieniamy wartości zmiennej sterujących wewnątrz pętli. Starajmy się dzielić program logicznie na podprogramy. Ułatwia to uruchomienie i późniejsze zmiany. Pisząc programy uważajmy aby w podprogramach nie wykorzystywać tych samych zmien-

nych, co w innych segmentach. Niezamierzona zmiana wartości zmiennych przez podprogram w innym segmencie to częste źródło trudnych do wykrycia błędów.

Zwróćmy uwagę, szczególnie przy stosowaniu instrukcji porównania IF, że liczby zmiennopozycyjne reprezentowane są w komputerze ze skończoną dokładnością. Znaczy to po prostu, że np. działanie $11,5 * 2,5$ nie musi dać wyniku dokładnie 26,25.

Efektywność działania programu starajmy się osiągnąć przez inteligentny dobór algorytmów i struktur danych, a nie przez trywialne chwytły w rodzaju eliminacji wyliczania indeksów przy korzystaniu z tablic. Pamiętajmy zawsze o tym, że dopóki program nie działa poprawnie, jego efektywność nie ma żadnego znaczenia. Szybki, lecz działający błędnie program jest bezużyteczny.

Starajmy się nie ulepszać algorytmów bez koniecznej potrzeby. Nie należy tego także traktować jako "sztuka dla sztuki". Program można ulepszać tylko wtedy, gdy występuje taka konieczność i wiemy dokładnie, która część programu wymaga udoskonalenia.

Nie poświęcajmy czytelności dla efektywności. Większość spotykanych sugestii odnośnie poprawy efektywności opiera się na trikach obniżających czytelność.

Pamiętajmy, że niektóre algorytmy (co widać szczególnie na przykładach algorytmów wyszukiwania i sortowania) są setki razy szybsze od innych. Niżej podam kilka sposobów poprawy efektywności programów. Stosujmy je z umiarem i tylko do tych fragmentów programu, które rzeczywiście tego wymagają.

W typowym programie większość czasu przypada na jego niewielką część (często 5%), z reguły są to wielokrotnie wykonywane pętle lub wielokrotnie wywoływane podprogramy. Tam właśnie dokonać można ulepszeń. Wprowadzanie ich w innych miejscach może dać niezauważalne skrócenie czasu wykonania przy kompletnym zaciemnieniu całego programu.

Pamiętajmy, że operacje matematyczne wykonywane są z różnymi szybkościami. Oto lista operacji arytmetycznych uszeregowanych od najszybszych do najwolniejszych.

1. Dodawanie lub odejmowanie
2. Mnożenie
3. Dzielenie
4. Potęgowanie

Widzimy, że dodawanie jest szybsze niż mnożenie, zatem mnożenie przez niewielkie liczby rzeczywiste można zastąpić wielokrotnym dodawaniem np. zamiast:

3*I

można zapisać:

I+I+I

W analogiczny sposób można podnosić do niewielkiej potęgi zastąpić wielokrotnym mnożeniem.

Można pozbywać się operacji także drogą przestawienia wyrażeń. Na przykład:

X=2*Y+(A-1)/P+2*T

można zamienić na:

X=2*(X+T)+(A-1)/P

```
1050 REM FOR I=DLSTART TO DLEND
1060 REM PRINT I,PEEK(I)
1070 REM NEXT I
1080 REM
1090 POKE 512,0
1100 POKE 513,6
1110 REM
1120 FOR I=1536 TO 1550
1130 READ A
1140 POKE I,A
1150 NEXT I
1160 REM
1170 FOR I=DLSTART+6 TO DLSTART+28
1180 POKE I,130
1190 NEXT I
1240 POKE 54286,192
2000 REM
2010 DATA 72
2020 DATA 173,11,212,141,10,212,141,24,2
08,141,26,208
2030 DATA 104,64
READY
```

eliminując jedno mnożenie. Dzielenie jest wolniejsze niż mnożenie, więc zamiast dzielić należy, jeżeli jest to możliwe, możemy mnożyć przez odwrotność. Zamiast dzielić przez 4 można mnożyć przez 0,25 itp. Jeżeli wielokrotnie dzielimy przez tę samą zmienną można zastosować zmienną roboczą, w której zapamiętujemy odwrotność i następnie mnożyć przez tę zmienną.

Użycie zmiennej roboczej może być także korzystne w wypadku obliczeń w pętlach. Na przykład zamiast:

```
115 FOR I=1 TO 1000
125 ALFA1=SIN(BETA)
135 ALFA2=SIN(BETA)
175 ALFA9=SIN(BETA)
185 NEXT I
```

Można zapisać:

```
115 POMOC=SIN(BETA)
125 FOR I=1 TO 1000
135 ALFA1=POMOC
175 ALFA9=POMOC
185 NEXT I
```

W pierwszym przypadku SIN(BETA) jest obliczany 9000 razy, w drugim tylko jeden raz - przed wejściem w pętlę.

Jeżeli pętli nie da się uniknąć pamiętajmy, aby wszystkie obliczenia, których wykonanie w pętli nie jest konieczne, umieścić przed pętlą. Niewielkie rachunki można często wykonywać w ogóle bez posługiwania się pętlami.

Przy optymalizacji pętli najwięcej uwagi starajmy się zawsze poświęcić pętli wewnętrznej. Oto kolejny przykład:

```
115 FOR A=1 TO 50
125 FOR B=1 TO 50
135 FOR C=1 TO 50
215 NEXT C
225 NEXT B
235 NEXT A
```

W pętli C każda operacja wykonywana jest 75000 razy. Bardzo niewielkie ulepszenie działań wewnątrz pętli C zapewni nam dużo większy zysk niż znaczne nawet zmiany w pętlach A oraz B.

Raz jeszcze przypomnijmy, że ulepszenia należy stosować tylko tam, gdzie jest to konieczne, czyli w wielokrotnie wykonywanych pętlach, lub wielokrotnie wywoływanych podprogramach. Poza tymi miejscami zupełnie się to nie opłaca.



Jeżeli konieczne jest zwiększenie szybkości działania naszego programu możemy to uzyskać przez:

- umieszczenie wielokrotnie wykonywanych pętli i często używanych podprogramów w liniach o jak najniższych numerach,

- umieszczenie pętli FOR-NEXT w jednej linii. Pętla taka wykonywana jest nieco szybciej niż umieszczona w kilku liniach.

Co natomiast można zrobić, jeżeli nasz program nie mieści się w pamięci? Jeżeli nie możemy zmniejszyć zajętości pamięci przez operacje typu zmiana algorytmu, ograniczenie wymiarów tablic, spróbujmy:

- przypisać często występujące stałe do zmiennych i w programie będziemy odwoływać się wówczas do odpowiednich zmiennych,

- używać podprogramów zamiast wielokrotnie występujących, takich samych linii programu,

- wykorzystać elementy tablic o zerowych indeksach np. X(0),

- zastosować instrukcje READ i DATA do inicjacji zmiennych, zamiast instrukcji podstawiania lub jeszcze lepiej - użyć INPUT lub GET oraz odpowiednich zbiorów,

- zastosować w jednej linii po kilka instrukcji,

- usunąć wszystkie linie komentarza (REM).

Pamiętajmy jednak, że powyższe zmiany mogą znacznie obniżyć czytelność programu. Ich stosowanie starajmy się ograniczyć do naprawdę koniecznych przypadków.

Na koniec jeszcze jedna uwaga. Znaczące zwiększenie szybkości działania programu i zmniejszenie zajmowanego obszaru pamięci możemy uzyskać stosując segmenty w kodzie maszynowym, co jednak jest sprzeczne z założeniem przejrzystości programu. W Atari Basic do wywołania procedur w kodzie maszynowym służy funkcja `USR`.

Nic wyczerpaliśmy tu oczywiście, ani wszystkich problemów związanych z tak zwanym dobrym stylem programowania, ani z poprawą szybkości działania programów. Zainteresowanych bliżej tym problemem zachęcam do lektury literatury fachowej. Ponadto zawsze należy zapoznać się szczegółowo z fabrycznymi instrukcjami do użytkowanych komputerów lub interpreterów, aby w pełni wykorzystać dawane przez nie możliwości.

Opracował: Łukasz Nowak

 **AMIGA.net.pl**
P r z e j d ź n a n a s z ą s t r o n ę

OFERUJEMY:

**Nową literaturę przeznaczoną dla użytkowników komputerów:
COMMODORE, AMSTRAD CPC, ZX SPECTRUM, ATARI, AMIGA**

POSZUKUJEMY:

**Autorów artykułów pragnących podzielić się wiedzą oraz
doświadczeniami z zakresu użytkowania mikrokomputerów.**

DOS XE

Wśród systemów dyskowych na nasz komputer, prym niewątpliwie wiedzie Sparta oraz MyDOS. Mało kto jednak wie, że swego czasu czasu i Atari wypuściło całkiem dobry DOS. Dlaczego nie zdobył on popularności?

DOS XE powstał w roku 1988 i był rozpowszechniany wraz ze stacją XF551. System ten jest bodajże jedynym, z którym XF pracuje nie sprawiając kłopotów. Stacja XF znana jest bowiem z tego, iż nie potrafi rozpoznać gęstości typu Double, jeśli uprzednio pracowała w Single lub Medium. Sprawia to wiele problemów w pracy, jednak nie podczas użytkowania DOS XE, gdyż tego typu kłopoty po prostu nie występują. I tu nasuwa się bardzo ważny wniosek, bo jeśli XE potrafi sprawnie kierować stacją, a inne DOSy nie, to czy na pewno całą winę w tym przypadku można zgonić na stację?

Przejdźmy jednak do tematu, czyli opisanie samego DOS XE. Jest on jednym bardziej udanych systemów. Posiada wiele funkcji, pozwala na zakładanie podkatalogów, a obsługa nie powinna sprawiać kłopotów nawet najbardziej zielonym użytkownikom.

Niestety, DOS XE stosuje swój własny sposób zarządzania plikami, stąd nie jest kompatybilny z resztą DOSów, jednak po części jest na to rada, ale o tym za chwilę. XE obsługuje wszystkie popularne gęstości, tzn. pojedynczą, średnią, podwójną i podwójną dwustronną. Mają one jednak trochę inne nazwy:

AT810 - czyli Single, 40 ścieżek o 18 sektorach po 128 bajtów, 90kB,

AT1050 - Medium (zwany czasem Dual lub Enhanced), 40 ścieżek o 26 sektorach po 128 bajtów, 130kB,

SSDD - Double, 40 ścieżek o 18 sektorach, każdy po 256 bajtów, 180 kB,

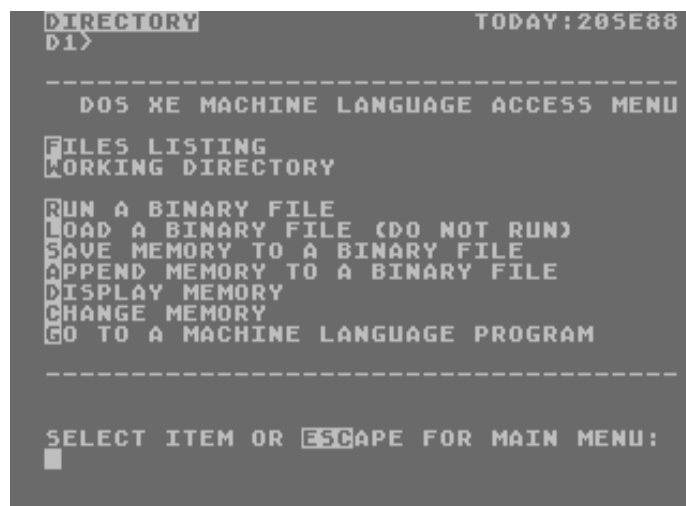
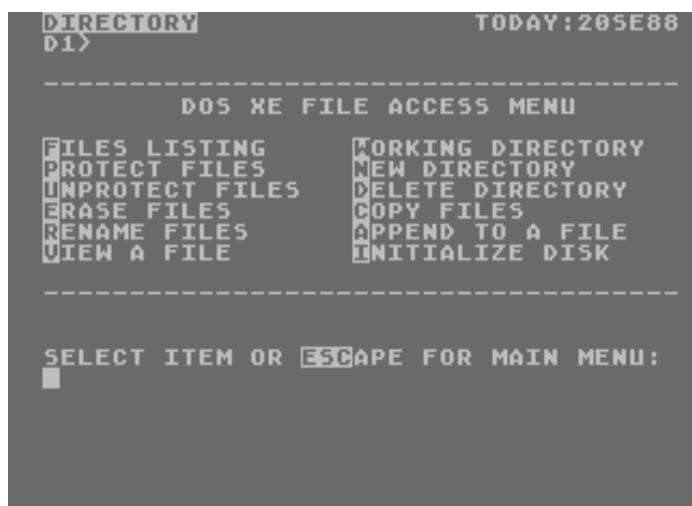
XF551 - Double dwustronny, 80 ścieżek (po 40 na stronę), o 18 sektorach po 256 bajtów, 360kB.

Jak widać nazwy formatów zostały nadane na podstawie stacji produkowanych przez Atari i uzyskiwanych przez nie możliwości. Oczywiście realna pojemność dyskietki, podana przy każdym rodzaju formatu, jest umowna. W rzeczywistości na ogół jest trochę mniejsza.

Zwróćmy uwagę także na obsługę. Po wczytaniu ukazuje się menu, z którego możemy wybrać:

- operacje na plikach,
- funkcje w języku maszynowym,
- specjalne funkcje DOSu.

Opcjonalnie istnieje możliwość przejścia do Basica lub cartridge'a. Wszystkie funkcje w DOS XE wybieramy wpisując jej pierwszą literę oraz zatwierdzając przez RETURN.





DOS XE - File Access Menu

Po wybraniu FILE ACCESS MENU, czyli menu operacji plikowych, ukazuje się kolejne menu. Są tu wszystkie podstawowe funkcje, takie jak: katalog, zabezpieczanie, odbezpieczanie, kasowanie, zmiana nazwy, przeglądanie zawartości plików, zmiana podkatalogu, tworzenie nowych podkatalogów i ich kasowanie, kopiowanie plików, dołączanie do już istniejących oraz formatowanie dyskietki.

Na uwagę zasługuje sposób przedstawiania przez DOS XE zawartości dysku. Obok nazwy pliku, znajdziemy tu także informacje o jego długości (w bajtach), datę powstania i datę wykonania ostatniej operacji. Ilość miejsca wolnego na dyskietce przedstawiona jest w kilobajtach, przy czym kilobajt w tym przypadku oznacza liczbę 1000 bajtów, a nie 1024.

DOS XE - Machine Language Access Menu

Druga z omawianych opcji to MACHINE LANGUAGE ACCESS MENU. Znajduje się tu również funkcja wyświetlenia katalogu i zmiana podkatalogu, a oprócz tego funkcja "Run Binary File", która zaraz po wczytaniu programu pozwala go uruchomić. Można również wczytać program bez jego uruchomienia, natomiast za pomocą funkcji "Save Memory" możemy zapisać dowolny fragment pamięci do postaci binarnej.

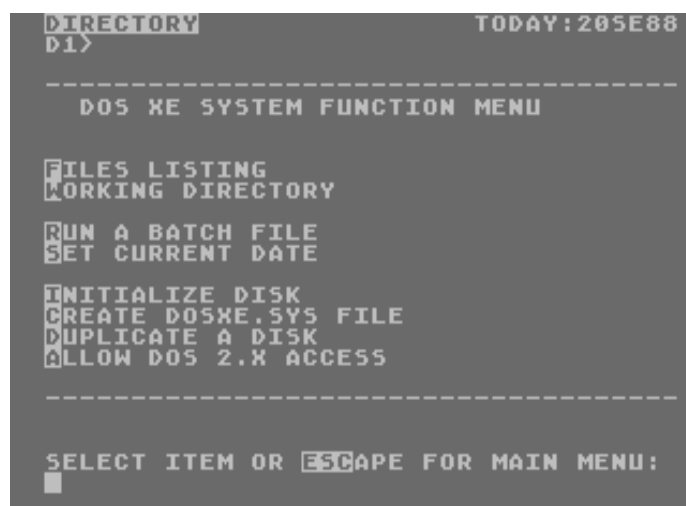
Za pomocą następnej funkcji możemy dodać fragment pamięci do już istniejącego pliku. Dzięki "Display Memory" obejrzymy fragment pamięci komputera. Kolejna opcja "Change Memory" - jak sama nazwa wskazuje - pozwala na modyfikację zawartości komórek pamięci.

Ostatnia w tym podmenu opcja uruchamia program maszynowy znajdujący się w pamięci, pod wskazanym wcześniej adresem. Sam adres możemy wpisać w postaci szesnastkowej lub dziesiętnej poprzedzonej znakiem krzyżyka.

DOS XE - System Function Menu

Trzecie - SYSTEM FUNCTION MENU - zawiera - podobnie, jak poprzednie - funkcję listy plików i zmiany podkatalogu. Poniżej istnieje opcja, dzięki której możemy wczytać i uruchomić plik typu BATCH, czyli wsadowy zawierający rozkazy DOSu, które są wykonywane kolejno przez system bez potrzeby ingerencji użytkownika. Opcja "Set Current Date" to nic innego, jak ustalanie nowej daty (w formacie DZIEŃ-MIESIĄC-ROK). Jest tu również funkcja formatowania i kopiowania całej dyskietki.

Wybór "Create DOSXE.SYS File" spowoduje zapisanie ustaleń dokonanych przy pomocy programu SETUP.COM. Program ten wczytać należy funkcją "Run Binary File". Natomiast wybór



funkcji "Allow DOS 2.x Access" spowoduje wczytanie programu DOS2.SYS.

Możemy teraz korzystać z programów zapisanych w formacie DOSu 2.0/2.5. Wczytujemy je jak zwykłe programy, z tą różnicą, że jako urządzenie podajemy "A:". W ten sposób można łatwo przekształcić program do formy akceptowalnej dla DOS XE. Wystarczy wybrać opcję kopiowania plików, załadować wybrany plik z urządzenia "A:", a potem zapisać na urządzeniu "D:", czyli na dyskietce z DOS XE. Niestety nie wszystkie programy po takiej czynności będą działały, ale warto spróbować. Na dyskietce systemowej znajduje się także ciekawy program COPY3_XE.COM. Jest to konwerter, który przekształca pliki z formatu DOSu 3.0 na format XE.

Tak przedstawiają się wszystkie ważne funkcje programu. Warto również dodać, że wydawanie rozkazów DOSowi można skrócić jeszcze bardziej. Przykładowo jeżeli chcemy załadować do pamięci program PRO.COM, w głównym menu wystarczy wpisać:

M L PRG.COM

Jak widać wpisujemy tylko pierwsze litery interesujących nas opcji, a spacja w tym przypadku zastępuje RETURN. Dzięki temu wszystkie funkcje DOS XE można uruchamiać dużo szybciej.

Opracował: Mariusz Wasilewski

Automatyczna numeracja

Numerowanie linii programu to ważny, ale jednocześnie bardzo często sprawiający kłopoty, element Basica. Proponuję więc zastanowić się nad tym, jak uzyskać automatyczne liczby obok poszczególnych linii bez komplikacji. Okazuje się, że nie jest to takie trudne nawet jeśli nie dysponujemy stacją dyskieta. Całość zapewnia wygodę podczas korzystania z Atari BASIC.

Prezentowany program dodaje automatyczne numerowanie linii do Atari BASIC. Jest on używany w bardzo podobny sposób jak funkcja Asemblera o nazwie EDITOR NUM. Działanie polega na stworzeniu programu startowego kasety, który ma zostać załadowany po włączeniu komputera w trybie Basic. Procedura może zostać wywołana w dowolnym momencie i może być łatwo odłączona lub ponownie aktywowana.

Wystarczy wpisać uważnie program, a następnie wykonać następującą instrukcję w trybie bezpośrednim:

A = USR(12288)

Usłyszymy dwa sygnały dźwiękowe, które oznaczają, że trzeba nacisnąć klawisze PLAY i RECORD na magnetofonie. Oczywiście w środku musi być kaseta. Dalej naciskamy klawisz RETURN. To wszystko.

Aby skorzystać z automatycznej numeracji trzeba teraz załadować nagrany program w zwykły sposób. Uruchamiamy więc komputer z wciśniętym klawiszem START, a gdy słychać sygnał dźwiękowy, naciskamy PLAY, a następnie RETURN.

Po chwili program znajdzie się w pamięci. Aby go aktywować, należy wykonać następujące czynności w trybie bezpośrednim:

A = USR(1550, a, b)

```
10 DIM A$(100), B$(2), H$(23) : H$ = "
(,) (A) (B) (C) (D) (E) (F) (G) (H) (I) ! ! ! ! ! (J) (K) (L) (
M) (N) (O)
"
20 MEM = 1536 : M = -1
30 READ A$
40 FOR I = 0 TO 49 : B$ = A$ (I*2 + 1, I*2 + 2)
50 IF B$ = " YY " THEN RESTORE 700 : MEM = 12
288 : M = -1 : GOTO 30
60 IF B$ = "ZZ" THEN ? "ALL DONE" : END
70 N = 0 : FOR J = 1 TO 2 : N = N * 16 + ASC
(H$(ASC(B$(J)) - 47)) : NEXT J : M = M + 1 :
POKE MEM + M, N
80 NEXT I
90 GOTO 30
100 DATA 000200060D06A93C8D02D318606068C900D00B
A90085CBA91085CC4C4F0685CFC901F00B20D306A5D585C
DA5D485CE20D306
200 DATA A5D585CBA5D485CCA5CFC902D00FF838A5CEE5
CC85CEA5CDE5CB85CDD8A9808510A9628D0802A9068D090
2A9C0851060A5CF
300 DATA C9FFF011AD09D2C90CF0034CBEFFA9FF85CF4C
BEFFAD09D2C90CD015A9808510A9BE8D0802A9FF8D0902A
9C085104CBEFFA9
400 DATA 0085CFF818A5CE65CC85CEA5CD65CB85CDD820
BD06A5CD20CA06A5CE20BD06A5CE20CA064CBEFF29F04A4
A4A4A18693020A4
500 DATA F660290F18693020A4F6606885CC6885CB6885
D56B85D420AAD9A5D4C941F00BA5D585D4A90085D54CF70
6A5D685D4A5CB48
600 DATA A5CC4860YY
700 DATA A210A9039D4203A9089D4A03A9089D4B03A94A
9D4403A9309D45032056E43028A90B9D4203A9009D4403A
9069D4503A9009D
800 DATA 4803A9019D49032056E4300AA90C9D42032056
E430006860433A9BZZ
```



gdzie "a" jest początkowym numerem linii, a "b" jest rozmiarem pojedynczego kroku (odstępu pomiędzy wartościami). Jeśli pominiemy "b", liczba "a" stanie się wielkością kroku. Jeśli zarówno "a", jak i "b" zostaną pominięte, rozpocznie się numerowanie rozpoczynając od 10, a kolejne linie będą miały wartości również powiększone do 10.

Aby aktywować numerowanie trzeba nacisnąć dwa razy klawisz RETURN. Aby wyłączyć automatyczną numerację trzeba zrobić to samo. Funkcja działa więc na zasadzie przełącznika.

Wpisanie całego programu może chwilę potrwać, ale to zrobić, szczególnie jeśli chcemy pisać dłuższe programy. Przy

dłuższych programach ułatwia orientację w napisanym kodzie. Może przeszkadzać właściwie tylko przy bardzo długich listingach i gruntownych przeróbkach rozbudowanego programu, ale wtedy programista z pewnością nie należy do początkujących i poradzi sobie bez względu na okoliczności.

Kasowanie linii w Basicu

Chyba każdy, kto próbował napisać długi program w Basicu na Atari, napotkał najpoważniejszy problem w systemie, czyli możliwość zawieszenia. Gdy w programie zostaną wprowadzone zmiany, komputer może zostać "zablokowany". W takich przypadkach zazwyczaj nie ma innego wyjścia jak tylko wyłączenie komputera i włączenie go ponownie. Niestety pamięć zostanie wtedy całkowicie wymazana. Na szczęście problem ten można rozwiązać.

Dzięki przedstawionemu listingowi można uniknąć blokady systemu. Procedura spowoduje skasowanie określonego obszaru programu, na przykład od linii 100 do 200, bez powodowania awarii systemu. Jej użycie pozwala więc zaoszczędzić czas na przepisywanie tego samego listingu. Użytkownik musi wpisać tylko początek i koniec obszaru, który ma zostać skasowany.

Sam program najkrótszy jak to tylko możliwe, aby pozostawić wolną na większą ilość pamięci komputera, która może być przeznaczona na główny program. Aby jej użyć należy, po wpisaniu, wyświetlić jej treść za pomocą instrukcji LIST. Następnie w dowolnym momencie można ją przywołać a pomocą linii:

GOTO 32600

W odpowiedzi na pytanie, wpisujemy początkowe i końcowe numery linii programu, które mają być usunięte. Gdy operacja zostanie zakończona możemy kontynuować pracę nad swoim programem.

Niezależnie od tego udogodnienia proponuję pamiętać o cyklicznym wykonywaniu kopii wpisywanego programu. Nikt nie lubi tracić efektów swojej pracy, ale nie zawsze będziemy w stanie przewidzieć zachowanie komputera. I tutaj przydaje się pomoc.

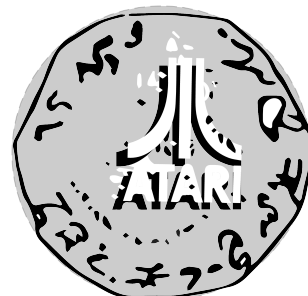
Opracował:

Łukasz Nowak

```
32600 GRAPHICS 0:? :? " "
32605 TRAP 32605:POKE 84,11:? "Start, Koniec";:INPUT 5, E
32610 IF INT(5)<> ABS(5) OR 5>32099 OR INT (E) <>ABS (E) OR
      E>32099 OR E<5 THEN ? CHR$(253): GOTO 32605
32615 GRAPHICS 0:? :?
32620 ? 5:5=5+1
32625 ? "CONT":POSITION 0,0:POKE 842, 13:STOP
32630 POKE 842,12:IF 5<=E THEN 32615
32635 GRAPHICS 0:END
```

WYKOPALISKA

małe i duże



CZARNY ORZEŁ

Jest to gra zręcznościowo-przygodowa wydana przez firmę Krysak. Z pozoru przypomina kolejną odmianę znanego hitu pt. "Operation Wolf", czyli strzelaniny z celownikiem, za pomocą którego możemy razić wrogów z karabinu i granatów. W miarę grania odkrywamy jednak, że Czarny Orzeł zawiera ona też elementy przygodówki, co ją bardzo uatrakcyjnia. Obsługa gry jest nieskomplikowana - w użyciu są jak zwykle: joystick, klawisz Spacji i szybki refleks.

Przed uruchomieniem rozgrywki dowiadujemy się, iż celem naszej akcji jest zlikwidowanie bazy rewolucjonistów, którzy wykradli najnowocześniejszy bojowy helikopter, będący bardzo groźną bronią do walki z przestępcami. Stąd pochodzi nazwa gry. Po udanej akcji pacyfikacyjnej mamy oczywiście obowiązek odstawić tę maszynkę do dowództwa.

Wcielamy się więc w rolę najlepiej wyszkolonego komandosa mającego ściśle określony plan akcji:

- przedostać się do bazy wroga,
- wysadzić bramę,
- zniszczyć zapasy paliwa i amunicji wroga,
- odnaleźć centrum dowodzenia rewolucjonistów,
- przejąć tytułowy helikopter.

Akcja rozpoczyna się na wyspie, przed bramą wrogiej twierdzy. Wieże strażnicze i bramę musimy zniszczyć samodzielnie przy pomocy granatów. Zasada

jest jedna - strzelamy do wszystkiego co się rusza, zwracając uwagę, że co jakiś czas trzeba pozbyć się faceta strzelającego z krzaków. Po zakończonej eksterminacji strażników, czas na właściwą rozgrywkę. Pojawi się strzałka, którą kierujemy na obrany kierunek - w tym wypadku do przodu. Na nowym ekranie znowu strzelamy do wszystkiego.

Dochodzimy do chaty i musimy "porozmawiać" z jej lokatorami, preferującymi ostrożność wobec obcych przejawiającą się gorącym powitaniem kulami karabinowymi. Nie należy pozostawiać im dłużny i po kolei likwidujemy drani. Teraz możemy już wejść do chaty, w której znajdujemy kanapkę znakomicie podwyższającą energię. Przy okazji zabieramy łom. Wychodzimy z chaty i szukamy składu amunicji. Tu oczywiście bezlitośnie wykorzystujemy możliwość uzupełni-

enia swego bojowego arsenału, przedtem jednak używamy łomu, aby otworzyć skrzynię.

Po wyjściu kierujemy się w tył i wchodzimy do tajnego biura rewolucjonistów, skąd zabieramy zapalki. Następnie idziemy dalej w tył i wysadzamy skład amunicji. Dalej wysadzamy skład paliwa, a następnie idziemy ponownie do biura rewolucjonistów. Wciskamy przycisk na ścianie - otworzy się tajne wejście, które doprowadzi nas do centrum dowodzenia nieprzyjaciela. Zabieramy helikopter "Czarny Orzeł". Gratulacje, misja zakończyła się sukcesem.

Podsumowując, gra nie jest trudna, wymaga tylko pewnej wprawy, bo na pewno nie przejdzie się jej od razu. Poziom trudności jest średni i na pewno nie zniechęci. Autora można pochwalić za to, że dodał do gry wątki z typowej





przygodówki, co ją urozmaiciło. Oprawa graficzna nie grzeszy urodą, ale i nie jest też najgorsza z możliwych - znowu poziom średni. Muzykę słycać na czołówce i na końcu, podczas gry są tylko efekty specjalne, co wcale nie przeszkadza, bo pozwala skupić się na samej istocie gry.

CZARNY ORZEŁ

Ocena:

grafika:



dźwięk:



grywalność:



banda Daltonów. Mieszkańcy Dawson czuli się przez nią zaszczerzeni. Niestety nikt nie dał Daltonom do tej pory rady i do tej pory dyktowali oni wszystkim swoje warunki. Kiedy wszelkie oczekiwania na pomoc zawiodły, do akcji wkraczasz Ty. Musisz przywrócić spokój i porządek. Masz także osobiste porachunki z groźną bandą, gdyż dochodzą pogłoski o zniknięciu Twojego starego przyjaciela. Tak więc Twoim celem w grze jest

odnalezienie i uratowanie od szubienicy starego przyjaciela oraz pokonanie Daltonów. Aby tego dokonać, musisz przebyć miasto poszukiwaczy złota, obóz indiański i fort, przy okazji przeżywać wiele przygód i narażając wielokrotnie swoje życie, na przykład przy grze w pokera.

GUNSLINGER

Gunslinger jest graficzno-tekstową grą przygodową w języku angielskim. Jako Kip Starr, były teksański strażnik, dzięki pomocy przewoźnika trafiamy do miasta Dawson City. Jest to czas, gdy Dzikie Zachód terroryzowała kilkuosobowa

Zwróćmy uwagę także na stronę techniczną gry. Przeważnie przygodówki na XL/XE przyzwyczaiły atarowskich graczy do tego, że ich obsługa była dość uciążliwa: trzeba wpisywać wszystkie słowa z klawiatury, uważając, czy nie ma w nich błędów i czy dane słowo będzie uwzględnione w słowniku. Jednym słowem: dużo nerwów przy obsłudze i mało wygody.

Jednak w grze "Gunslinger" jest inaczej. Można ją obsługiwać z klawiatury lub joysticka, całkowicie zamieniając. Nie trzeba szukać wyrazów i wpisywać wszystkiego z klawiatury.

Ekran jest podzielony na kilka części, z których dolna służy do wpisywania komend lub odczytywania komentarzy. Powyżej jest małe okienko z opisem miejsca oraz ze wskaźnikami do sterowania. Na górze widzimy okno z kolorową grafiką przedstawiającą grafikę ilustrującą akcję.



Po prawej stronie widoczna jest okno ze spisem wszystkich słów dostępnych w grze. Można tutaj wybierać między rzeczownikami (Nouns), czasownikami (Verb) lub przymkami (Preps) i kliknąć dwa razy strzałką na wyraz, aby dane słówko przenieść na dolny ekran z kometkami. Po wybraniu konkretnego wyrazu można go zatwierdzić przez kliknięcie na "CR", albo wciśnięcie klawisza RETURN.

Oczywiście, wszystko można wpisywać także z klawiatury. W grze jest sporo wątków, które potrafią rozbawić, natomiast zagadki są na poziomie umiarkowanym. Autorzy zrezygnowali z dodatkowych utrudnień, tak więc czas jest nielimitowany. Jednocześnie nie ma podpowiedzi, tak jak w starszych produkcjach. Wpisanie "HELP" zawsze wyświetli napis:

Those who help themselves.

Widać więc, że grający jest zdany wyłącznie na siebie, ale to nie rzutuje na całą grę, gdyż, jak już wspomniałem, zagadki nie są trudne. Za to rozmiary gry są imponujące. Trzy pełne dyskiety gwarantują sporo zabawy i czasu. Nie ma mowy, aby ktoś przeszedł tę grę od razu, bez prób czy opisów. Urozmaicona sceneria znakomicie wpływa na grywalność. Ogólnie jest to jedna z niewielu przygodówek napisanych z takim rozmachem.

Pozostałe rzeczy, takie jak grafika czy możliwość nagrania stanu gry to standard i nie brakuje tego także w Gunslingerze. Niestety nie ma dźwięku, ale nie jest on też konieczny. Mała uwaga: stan gry trzeba nagrywać na czystym, sformatowanym dysku w gęstości Single, inaczej operacja nie będzie



możliwa do wykonania. Na koniec garść porad na drogę: prawie wszystkie znalezione obiekty należy obejrzeć (look) albo zbadać (examine) i zabrać (get). Należy rozmawiać z napotkanymi osobami - może się to okazać wielce pomocne. Warto często nagrywać stan gry, aby nie wracać wciąż na sam początek. Wykonujemy to za pomocą komendy Save, dalej wkładamy do stacji czystą dyskietkę i podajemy literę jako identyfikator, ale pomijając "A". Polecenie Restore Game odczyta z później tego dysku stan gry.

Gracz jest sprawiedliwym, samotnym wojownikiem - postacią pozytywną, dlatego w zasadzie nie można stosować przemocy. Próba pobicia fryzjera skończy się ostrzeżeniem o następującej treści:

As an ex-ranger, your moral fiber is much too strong to break the law... without a reason.

czyli:

Twoja moralność, jako byłego żołnierza, jest

zbyt silna, aby łamać prawo... bez powodu.

Od razu wiemy, co autor miał na myśli. Polecenia można skracać, na przykład wpisując:

- GAM zamiast GAMBLE,
- INV zamiast INVENTORY
- GIV TIC zamiast GIVE TICKET.

Kierunki poruszania są skrótami angielskich nazw - i tak:

- N oznacza North (Północ),
- S oznacza South (Południe),

GUNSLINGER

Ocena:

grafika:



dźwięk:



grywalność:





- W oznacza West (Zachód),
- E oznacza East (Wschód).

Z pewnością nie każdy ma na tyle dużo samozaparcia, aby przejść grę bez żadnej pomocy. Możliwe, że przydałoby się małe ułatwienie dla wszystkich, którzy nie ukończyli pomyślnie Gunslingera i utknęli albo zobaczyli bardzo miły napis: "Perils of the West overcome you", czyli "Uległeś niebezpieczeństwu Zachodu".

Uważam jednak, że samodzielne odkrywanie sekretów gry jest największą radością, a niewygórowany poziom trudności spowoduje, że wiele osób zainteresuje się fabułą. W końcu jesteśmy przyzwyczajeni do rozwiązywania zagadek, prawda?

THE PAWN

Pawn jest tekstową przygodówką zawierającą grafikę, która ma ilustrować akcję. Podobnie jak w większości takich gier, bierzesz rolę poszukiwacza przygód, spotykając się z innymi postaciami i zastanawiając się, co robić dalej? Wystarczy wpisać polecenie, używając komend takich jak "Look in the fontanna" czy "Drop all but the pot and arow". Podobnie jak w większości

podobnych tytułów, gra jest wypełniona intrygującymi zagadkami, które jednak są dość typowe. Jeśli chodzi o obsługę, jest tu kilka ciekawych funkcji. Zastosowany algorytm jest dobry, umożliwia wykonywanie rozmów z różnymi postaciami i pozwala na szeroki zakres działań.

Gra posiada stronę wizualną, która w dużej części naprawdę robi wrażenie. W wersji ST cieniowanie wykonane jest znakomicie. Obrazy trawiastej równiny i pustyni, z ich trójwymiarową perspektywą i kompozycją, są warte obejrzenia. Moimi dwoma ulubionymi są kamienny most i pałacowe ogrody. Ten ostatni używa profesjonalnego cieniowania i tekstur, aby uzyskać naprawdę doskonały efekt.

Opracował:

Mariusz Wasilewski

THE PAWN

Ocena:

grafika:



dźwięk:



grywalność:



ACTION!

Jak pisać programy?

Język Action! powstał w roku 1983 w amerykańskiej firmie Optimized Systems Software. Jest to język programowania zaprojektowany specjalnie dla ośmiobitowych komputerów Atari i będący językiem pośrednim między Pascalem i C.

System Action umieszczony jest na cartridge'u, który zawiera monitor, edytor i kompilator. Ponadto dostępna jest dyskietka "Action! Tool Kit" zawierająca dodatkowe procedury, które można dołączyć do własnego programu.

Action! nie ma swojego odpowiednika dla innych komputerów. Uruchamia się automatycznie po włączeniu komputera. Jeśli w zestawie znajduje się stacja dysków, to ze znajdujących się w niej dyskietki odczytywany jest jeszcze DOS. Po uruchomieniu zgłasza się edytor Action. Programy napisane w Action mogą być uruchamiane (nawet po kompilacji) tylko wtedy, gdy cartridge znajduje się w porcie. Istnieje jednak możliwość ich uruchomienia samodzielnie po dołączeniu pliku runtime.

Cartridge Action zajmuje obszar pamięci od adresu 40960 (\$A000) do 49151 (\$BFFF). Poniżej znajduje się pamięć obrazu. Cała pamięć operacyjna od szczytu DOSu (lub od adresu 1792, gdy nie ma DOSu) do początku pamięci obrazu jest przeznaczona dla systemu. Do dyspozycji użytkownika pozostaje więc tylko szósta strona pamięci RAM (1536-1791 = \$0600-\$06FF).

Duża szybkość wykonywania programu czyni zbędne stosowanie procedur w języku maszynowym, a konieczne, stosunkowo niewielkie fragmenty kodu maszynowego, można dołączać bezpośrednio do treści programu albo umieszczać w tablicach znakowych o dowolnym lub ustalonym położeniu w pamięci.

Sercem Action! jest monitor. Zarządza on pracą pozostałych części systemu: edytora, kompilatora i biblioteki. Dwa ostatnie elementy są bezpośrednio niedostępne dla użytkownika i korzysta się z nich poprzez monitor, natomiast edytor wykorzystuje się bezpośrednio

po wywołaniu go z monitora. Kompilator normalnie nie rozróżnia małych i dużych, co pozwala na dowolne wpisywanie treści programu. Po uruchomieniu edytor zgłasza się automatycznie, a w ostatnim wierszu ekranu pojawia się napis "ACTION! (c) 1983 ACS". Wiersz ten służy do wyświetlania komunikatów i wpisywania poleceń redakcyjnych dla edytora. W przypadku równoczesnego korzystania z dwóch okien, wiersz komunikatów znajduje się pomiędzy nimi.

Edytor Action jest zwykłym edytorem tekstowym i jako taki posiada wiele specjalnych funkcji redakcyjnych.





Większość z nich uzyskujemy poprzez równoczesne naciśnięcie innych klawiszy oraz CONTROL i SHIFT. Niektóre dostępne są przy naciśnięciu CONTROL z innymi klawiszami. Niemal wszystkie funkcje edycyjne powodują wyświetlenie pomocniczych informacji w wierszu komunikatów, a także wymagają wprowadzenia odpowiednich danych. Po przejściu do monitora Action, w górnym wierszu ekranu pojawia się znak: ">".

Wiersz ten służy do wprowadzania poleceń, a pozostała część ekranu stanowi obszar wyświetlania komunikatów. Monitor pozwala na sterowanie pozostałymi elementami systemu. Służące do tego celu polecenia wpisuje się jako jednoliterowe symbole z odpowiednimi parametrami i zatwierdza klawiszem RETURN.

Struktura programu w języku Action zbliżona jest do programu w Pascalu, lecz ma większą elastyczność. Pod tym względem nasz język jest znacznie bliższy C. Każdy program w Action składa się z definicji zmiennych, procedur i funkcji. Poszczególne wiersze programu nie są numerowane. Instrukcje są oddzielane od siebie spacjami. Liczba instrukcji w wierszu jest dowolna, lecz długość wiersza nie może przekraczać 240 znaków. Podział programu na wiersze nie ma znaczenia dla kompilatora, lecz służy wyłącznie do poprawienia czytelności programu.

Każda zmienna, funkcja i procedura musi być definiowana przez użyciem, stąd ważna jest kolejność umieszczania deklaracji. Definicje te nie mogą być zagnieżdżane, czyli nie mogą znajdować się wewnątrz innych definicji.

Wykonywanie programu rozpoczyna się zawsze od ostatniej procedury lub funkcji zawartej w programie. Nazwy wszystkich procedur i funkcji mogą być

nawias, czyli "()". Ponadto nazwa funkcji musi być poprzedzona deklaracją typu wartości zwracanej przez funkcję. Definicja procedury lub funkcji kończy się słowem "RETURN".

```
PROC Main()
  BYTE CONSOL=$D01F ; Console reg.

  WHILE CONSOL>6 ; no console key?
  DO
    PrintE("Hello WIKIPEDIA!")
  OD

  RETURN
```

ACTION! (C)1983 ACS

W Action mogą występować tylko instrukcje proste, czyli polecenia wykonania pojedynczych operacji. Gdy zachodzi konieczność użycia instrukcji złożonej, a więc zawierającej kilka instrukcji prostych, trzeba ją zdefiniować w postaci procedury. Komentarze mogą być umieszczane w dowolnym miejscu programu i są oznaczane średnikiem.

Wszystko, co znajduje się w wierszu programu między średnikiem i końcem wiersza, jest ignorowane przez kompilator. Komentarze mogą więc zawierać dowolne znaki.

Action umożliwia łatwe korzystanie z procedur systemu operacyjnego oraz proste dołączanie do programów krótkich procedur w języku maszynowym. Dostęp do procedur systemowych i innych maszynowych uzyskuje się przez określenie adresu w definicji procedury. Natomiast krótkie fragmenty kodu maszynowego umieszcza się w dowolnych miejscach programu w postaci bloków kodu.

Blok kodu są to zamknięte w nawiasach kwadratowych ciągi wartości, które mogą być dowolnymi

stałymi, zmiennymi. Kompilator po natopkaniu bloku kodu przenosi zawarte w nim wartości bezpośrednio do tworzonego programu (kodu) wynikowego.

```
>
Initial Values
Byte: 255
Card: 32768 (0,128)
Int+: 1
Int-: -1
PutCD: 32768 (L=0,M=128)
PutID: 1
PutCD: 1 (L=1,M=0)
PutID: -1 (N=1,X=65535)
PutCD: 65535 (L=255,M=255)
Reset Values
Byte: 0
Card: 0 (0,0)
Int+: 0
Int-: 0
GetID: 1
GetID: 65535 (X=1,N=-1)
Read Values
Byte: 255
Card: 32768 (0,128)
Int+: 1
Int-: -1
```

```
Graphics Utilities Library for
ACTION!
-----
For years we've been touting the
speed of ACTION! and extolling its
fitness for graphics applications.
Now we're backing up our claims with
hard evidence--a set of graphics
routines written entirely in ACTION!.
With the Graphics Utilities Library
(GUL) you can add those special
touches to your ACTION! application
without spending a lot of time
developing your own graphics core.

So just what does GUL provide? If
it's advanced text-plotting you want,
GUL's got the answer: with its text
handling routines you can personalize
your text using a variety of
mix-and-match options like bold,
SPACE to continue or ESC to quit.
```

dowolne. Jednak po nazwie należy umieścić w nawiasie nazwy przekazywanych parametrów wraz z ich deklaracjami. Jeśli nie są wymagane żadne parametry, po nazwie procedury lub funkcji umieszcza się tylko pusty



Usprawniony katalog

Gdy używamy tradycyjnej stacji dyskieta, warto wiedzieć ile pozostało wolnego miejsca na każdym nośniku. Można robić to ręcznie, ale przecież powinien to robić komputer. Możemy uruchomić program FID, ale nie jest to zbyt wygodne. Wiele osób używa polecenia CATALOG w swoich programach, dlatego pokuszę się o jego wykorzystanie. Spróbujemy usprawnić jego działanie tak, aby wyświetlana była większa ilość informacji.

Katalog dyskiety znajduje się na ścieżce o numerze \$11. Zawiera 15 sektorów z informacjami o zawartości nośnika, a w każdym z nich zapisane są nazwy i objętości 7 programów. Mój program odczytuje sektory od \$F i umieszcza je w pamięci rozpoczynając od adresu \$2000. Sektor jest sprawdzany, a następnie odczytywany jest kolejny. Wpisy programów są sprawdzane pod kątem dwóch rzeczy. Pierwszym z nich jest odczytanie informacji, czy program nie został skasowany.

Wpis takiego programu nie jest usuwany z katalogu, ale zamiast tego pierwszy bajt na listingu będzie miał wartość \$FF. Jeśli natomiast program jest dostępny, dodawany jest bit zawierający długość. Bajt o największej

wartości jest przechowywany pod adresem \$6074 lub 24692. Po sprawdzeniu wszystkich siedmiu pozycji wczytywany jest kolejny sektor i proces ten trwa aż do zakończenia odczytywania wszystkich piętnastu sektorów. Wtedy powracamy do Basica.

Reszta programu to polecenia POKE i PEEK, które dotyczą dwóch miejsc zawierających sumę używanych sektorów. Najbardziej znaczący bajt jest pomnożony przez 256 i dodawany. Wyświetlana jest suma, a następnie odejmowana od 496, aby uzyskać

ilość pozostałego miejsca. Program odczytuje linie DATA i wprowadza liczby zaczynając od adresu \$6000. Wyświetla katalog, a następnie podaje wyniki obliczeń. Można go zastąpić programem z bardziej rozbudowanymi elementami, jak nazwa lub data, którą należy umieścić przez lub po linii 20, albo po linii 110. Jeżeli chcemy używać Basica do poważniejszych zastosowań niż uruchamianie prostych gier, poniższy program z pewnością przyda się na co dzień.

Mariusz Wasilewski

```
10 D$ = CHR$(4)
20 PRINT D$; " KATALOG "
30 FOR K = 24576 TO 24694
40 READ A
50 POKE K, A
60 NEXT
70 CALL 24576
80 X = PEEK(24692) : Y = PEEK(24693)
90 Z = X + 256 * Y
100 PRINT : PRINT "SEKTORY UZYWANE: " ; Z
110 PRINT "SEKTORY WOLNE: " ; 496 - Z
120 END
130 DATA 169, 96, 160, 76, 32, 217, 3, 173, 11, 32
140 DATA 201, 255, 240, 3, 32, 99, 96, 23, 8, 118, 96
150 DATA 173, 118, 96, 201, 7, 208, 28, 1, 69, 0, 141
160 DATA 118, 96, 169, 44, 141, 104, 96, 169, 11, 141
170 DATA 8, 96, 173, 2, 32, 201, 0, 240, 25, 141
180 DATA 81, 96, 76, 0, 96, 173, 104, 96, 105, 35
190 DATA 141, 104, 96, 173, 8, 96, 105, 35, 141, 8
200 DATA 96, 76, 7, 96, 96, 0, 1, 96, 1, 0
210 DATA 17, 15, 94, 96, 0, 32, 0, 0, 1, 0
220 DATA 0, 96, 1, 0, 0, 1, 239, 216, 0, 24
230 DATA 173, 116, 96, 109, 44, 32, 141, 116, 96, 144
240 DATA 3, 238, 117, 96, 96, 0, 0, 0, 0
```



Rysowanie w Basicu

Dialekt Applesoft Basic zawiera dwa podstawowe tryby graficzne - o niskiej rozdzielczości (Lores) oraz wysokiej rozdzielczości (Hires). Każdy z nich pozwala wyświetlać grafikę na pełnym ekranie lub jego części. W takiej sytuacji górna część ekranu zawiera grafikę, a dolna część może zawierać tekst. Tak działa zasadniczo instrukcja PRINT, gdzie dolny fragment ekranu działa w trybie tekstowym. O grafice w Basicu warto powiedzieć więcej, aby wykorzystać wszystkie możliwości komputera. Przyjrzyjmy się specyficznym instrukcjom, dzięki którym możemy używać różnych trybów graficznych, a także rysować linie.

Instrukcja GR ustawia tryb grafiki o niskiej rozdzielczości. Dostępny jest tryb 40 x 40 pikseli oraz pełnoekranowy 40 x 48 pikseli, oba obsługują 16 kolorów. Kolejne instrukcje HGR i HGR2 - to możliwość użycia trybu wysokiej rozdzielczości. HGR ustawia mieszany tryb, czyli 280 x 160 pikseli, natomiast HGR2 to tryb grafiki na pełnym ekranie, a więc rozdzielczość 280x192 pikseli. Tryby Hires mają tylko osiem kolorów.

Podobnie jak w trybach tekstowych, współrzędne w trybach graficznych zaczynają się w lewym górnym rogu ekranu (TOP LEFT, czyli współrzędne 0,0). Kiedy poruszamy się "w prawo", zwiększa się współrzędna X, a gdy poruszamy się "w dół" zwiększa się wartość Y. Jest to więc typowa zasada, choć nie na każdym komputerze.

Aby przełączać się z trybu mieszanego (GR lub HGR) do pełnoekranowego należy użyć linii:

POKE 49234,0

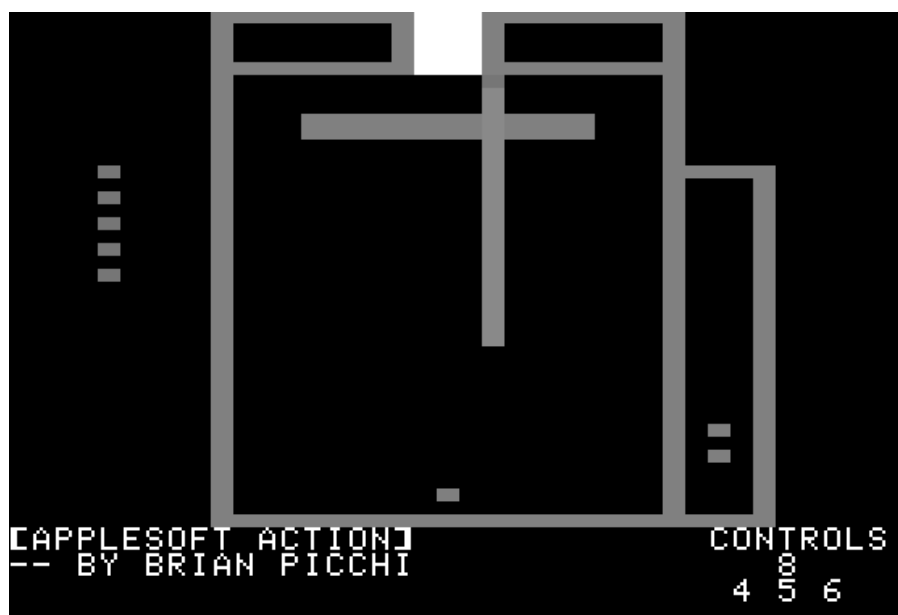
Wspomniane tryby GR i HGR nazywam mieszanymi, bowiem znajdują się na nich cztery linie trybu tekstowego - na dole ekranu. Aby wyświetlać na nich tekst, trzeba

ustawić kursor na jednej z czterech dolnych linii, na przykład za pomocą linii:

VTAB 21

Instrukcje GR, HGR, GGR2 czyszczą ekran na czarno, podobnie jak HOME w trybie tekstowym. Możemy oczywiście korzystać z różnych kolorów. Aby ustawić kolor jako bieżący należy użyć wyrażenia COLOR = N, gdzie "N" przyjmuje wartości od 0 do 15. Możemy wykonać podobną operację w trybie wysokiej rozdzielczości, ale korzystamy z wyrażenia HCOLOR = N, a tym razem "N" może być liczbą z zakresu od 0 do 7.

Rysowanie grafiki możemy zacząć od pojedynczych pikseli, co zrobimy za pomocą instrukcji PLOT w trybie





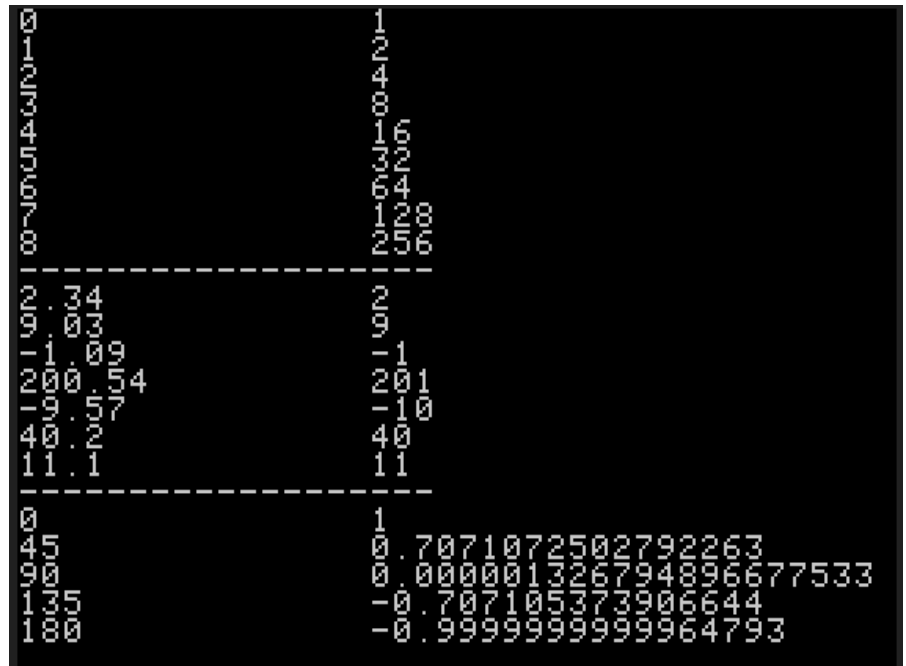
niskiej rozdzielczości i HPlot w trybie Hires. W obu przypadkach należy podać wartości X i Y, które są współrzędnymi piksela. Prawidłowe współrzędne w trybie Lores przyjmują wartości od 0 do 39, natomiast współrzędna pionowa Y jest rozszerzona w trybie pełnoekranowym i można stosować liczby od 0 do 47. Tryb wysokiej rozdzielczości charakteryzuje się wartościami X od 0 do 279, Y od 0 do 179 lub do 159 - w trybie mieszanym HGR.

Dalej możemy wyrysować linie pionowe lub poziome, co uzyskamy przy użyciu instrukcji HLine i VLine. Dotyczy to trybu niskiej rozdzielczości.

W trybach Hires linie pomiędzy dwoma punktami narysujemy za pomocą instrukcji HPlot, której trzeba podać współrzędne początkowe X, Y, słowo TO i współrzędne końcowe X2, Y2. Można również skorzystać z ostatniego zapamiętanego punktu, w tym przypadku podajemy tylko współrzędne końcowe po słowie TO, czyli:

HPlot TO X1, Y1

Program numer 1 zamieszczony w ramce rysuje złożony trójwymiarowy kształt geometryczny. Zauważmy, że jest to wykonywane przy użyciu wyłącznie prostych linii oraz podstawowych funkcji trygonometrycznych COS i SIN, które można używać w zwykły sposób wpisując wartość w nawiasie. Jak widać Basic może



służyć to bardziej skomplikowanych celów, mimo jego prostoty i teoretycznie dużym ograniczeniom.

Przyjrzymy się temu listingowi w działaniu. W linii 10 zaczynamy od instrukcji HOME, aby wyczyścić zawartość ekranu. Nie jest to zupełnie konieczne, ale moim zdaniem lepiej, aby nasza grafika była rysowana na pustym ekranie, szczególnie jeśli wcześniej używaliśmy trybu tekstowego.

Następnie wywołujemy HGR2, aby zainicjować tryb pełnoekranowy typu Hires z rozdzielczością 280 x 192 pikseli (HGR2 również czyści ekran). Za pomocą instrukcji HCOLOR ustawiamy bieżący kolor na 1 (czyli zielony). Posłuży on nam do rysow-

ania pikseli za pomocą linii zawierających dwie instrukcje - PLOT oraz HPlot.

Jeśli nie ustalimy bieżącego koloru, będzie domyślnie czarny i nie będziemy mogli zobaczyć efektów pracy programu. W liniach od 30 do 90 używamy prostej pętli FOR...TO...NEXT, dzięki której narysujemy kolejne proste linie - tutaj należy zwrócić uwagę na linię 80 z instrukcją HPlot.

Pełnoekranowy tryb niskiej rozdzielczości ma rozdzielczość 40 na 48 pikseli i obsługuje 16 kolorów. Program numer 2 pozwala zobaczyć jak wygląda to w praktyce.

Linia 10 ustawia tryb niskiej rozdzielczości za pomocą instrukcji GR, a następnie nakazuje Basicowi działać pełnoekranowo (POKE 49234,0). Linia 20 uruchamia pętlę FOR używając zmiennej C oznaczającej kolor. Zmieni się ona w zakresie od 0 do 15. Dopelniająca instrukcja NEXT znajduje się w linii 120. Linia 30 ustawia aktualny kolor na C.

W trybie niskiej rozdzielczości mamy 48 współrzędnych pionowych oraz 16 kolorów. Linia 40 tworzy nową pętlę

```
10 HOME : HGR2 : HCOLOR=1
20 PI = 3.14159 : D = 1.95
30 FOR A=0 TO PI*2 STEP PI/30
40 X1 = INT(280/2) + cos(A)*40 + 0.5
50 Y1 = 30 + sin(A)*20 + 0.5
60 X2 = INT(280/2) + cos(A + D) * 50 + 0.5
70 Y2 = 140 + sin(A + D)*40 + 0.5
80 HPlot X1,Y1 TO X2,Y2
90 NEXT
95 END
```

Program nr 1.



FOR, której zakończenie NEXT umieszczone jest w linii o numerze 60.

W tej drugiej pętli używana jest zmienna Y, dzięki której program narysuje trzy poziome linie - przy użyciu polecenia HLIN w linii 50, używając aktualnie wybranego koloru (czyli C). Linia 80 rozpoczyna kolejną pętlę, aby narysować poszczególne piksele ze wszystkich 16 kolorów wzdłuż poziomej linii.

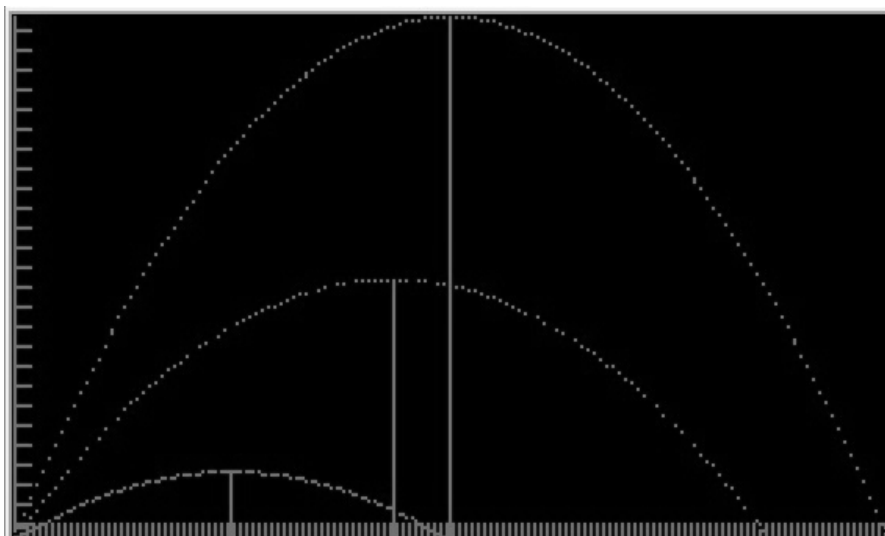
Używamy tu pętli FOR ze zmienną X, która służy do ustawiania bieżącego koloru, jak i do sterowania współrzędnymi piksela. Ostatnia pętla FOR ma swoje zakończenie w linii 110.

Z uzyskanego obrazu widać, że tryb GR rzeczywiście obsługuje 16 różnych kolorów, z wyjątkiem szarych 6 i 11, które wyglądają tak samo. Ze względu na użytą rozdzielczość, rozmiar pojedynczego piksela jest bardzo duży, piksele nie są również kwadratowe, bowiem ich szerokość jest niemal dwukrotnie większa niż wysokość.

Tryb niskiej rozdzielczości ma dwie podstawowe zalety - udostępnia dwa razy więcej kolorów i ma możliwość szybkiego wypełniania prostokątnych obszarów na ekranie. Niestety tracimy na jakości obrazu, poza tym nie mamy funkcji pozwalającej

```
10 GR : POKE 49234,0
20 FOR C = 0 TO 15
30 COLOR=C
40 FOR Y = C * 3 TO (C * 3 + 2)
50 HLIN 0, 39 at Y
60 NEXT
80 FOR X = 0 TO 15
90 COLOR=X
100 PLOT X * 2 + 4, C*3+1
110 NEXT X
120 NEXT C
130 END
```

Program nr 2.



narysować linię między dwoma dowolnymi punktami. Instrukcje HLIN i VLI są przeznaczone wyłącznie do rysowania linii poziomej lub pionowej. Mimo to możemy uzyskać szereg

ciekawych efektów na ekranie. Jak zawsze polecam samodzielne eksperymenty.

Opracował: Mariusz Wasilewski

Jaką największą wartość może mieć bajt?

Aby odpowiedzieć na to pytanie wszystkie bity ustawiamy jako "1", największą liczbą będzie więc 11111111. Teraz musimy tylko dodać wszystkie wartości mnożnika:

$$1+2+4+8+16+32+64+128 = 255$$

Ile wartości może mieć bajt?

Bajt może mieć wartości z zakresu od 0 do 255, czyli 256. Pamiętajmy, że zero też jest wartością.

Jak rozumieć zapisane wartości?

Wartości te można interpretować bezpośrednio, jako liczby, jednak można je również interpretować jako tekst, czas, tabelę itp. Wszystko zależy od tego w jakim kontekście bajt został użyty przez programistę.



Gry dla Apple II

Dla komputera Apple II nie powstało zbyt wiele gier, szczególnie jeśli porównamy ich ilość z biblioteką dostępną dla C64 czy Atari. Postanowiłem jednak pokazać, że i ten sprzęt może być platformą na długie jesienno-zimowe wieczory. Wybrałem gry, które nie są powszechnie znane z innych komputerów, a więc może zainteresuje się nimi więcej osób.

Akalabeth

Gdy pierwszy raz zobaczyłem ten tytuł, nic nie wiedząc o grze, moim pierwszym skojarzeniem była nazwa pochodząca z dalekiego wschodu. Okazało się, że mam do czynienia z produkcją z roku 1980 i była to jedna z pierwszych komercyjnych gier fabularnych. Co ciekawe, została napisana jeszcze w 1979 r. przez kilkunastoletniego Richarda Garriotta w Basicu, w zasadzie dla zabawy. Autor przez krótki czas rozprawdzał ją sam, w plastikowych torbach, a niedługo później za publikację zdecydowała się firma California Pacific Computer Company.

Pełna nazwa gry brzmi "Akalabeth: World of Doom". Autor inspirował się podobno dziełami J.R.R. Tolkiena, choć osobiście nie widzę większych nawiązań w fabule poza tytułem wy-

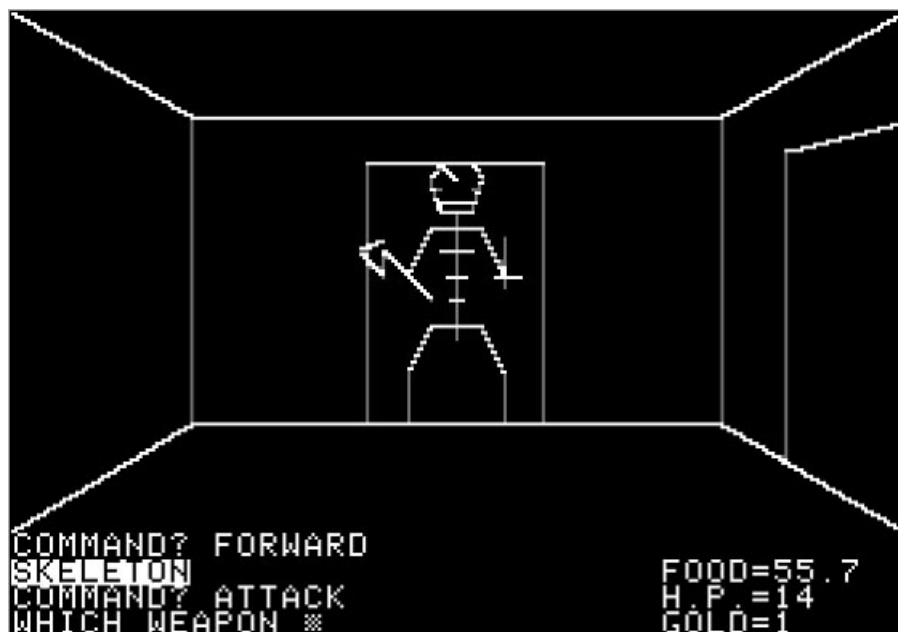
wodzącym się od Akallabêth, jednej z części Silmarillionu. Gracz ma za zadanie pokonanie dziesięciu potworów, a większość rozgrywki odbywa się w podziemiach, opartych wizualnie na prostej grafice wektorowej. Naprawdę ciekawe jest to, że fabuła widziana jest z perspektywy pierwszej osoby. Ponadto w grze pojawia się prosta mapa świata oraz wiele komunikatów tekstowych. Gra osiągnęła znaczną popularność, razem sprzedano kilkadziesiąt tysięcy kopii.

Eagle's Nest

Jest to dla mnie dużo ciekawszy tytuł, być może ze względu na bardziej dynamiczną akcję. W tym przypadku

mamy do czynienia ze strzelaniną wydaną przez firmę Mindscape, rozgrywająca się w realiach II wojny światowej. Gra miała swoje wersje na wielu różnych platformach sprzętowych, od 8-bitowego Atari, Amstrada CPC, ZX Spectrum, Commodore 64, Amigę i Atari ST uzyskując bardzo pozytywne recenzje.

Gracz wciela się w komandosa walczącego po stronie aliantów, jego zadaniem jest uwolnienie zakładników oraz wysadzenie w powietrze tytułowej nazistowskiej twierdzy – Orlego Gniazda. Na początku można wybrać poziom trudności oraz określić misję do wykonania. Tym razem nasza postać widoczna jest z góry. Chodzimy po labiryncie, można





także poruszać się windą pomiędzy kilkoma piętrami budynku. Na swojej drodze napotykamy przeciwników, czyli niemieckich żołnierzy, których oczywiście trzeba eliminować. Gracz może uzupełniać ilość amunicji oraz leczyć poziom otrzymanych obrażeń przy pomocy znajdujących przedmiotów. Mogą to być skrzynki z amunicją, jedzenie, zestawy pierwszej pomocy i tym podobne rzeczy.

Poruszanie się utrudniają drzwi, których część można otworzyć przy użyciu amunicji, do otwarcia innych niezbędne jest znalezienie kluczy. Na mapie rozmieszczone są także ładunki wybuchowe, których trafienie kończy grę. W celu uratowania zakładnika musimy doprowadzić go do wyjścia, a w celu wysadzenia twierdzy trzeba uruchomić detonatory zlokalizowane w różnych częściach budynku.

Mystery House

Kolejną grą, o której chcę opowiedzieć jest "Mystery House", która pod względem wizualnym bardzo przypomina "Akalabeth". Znowu mamy prostą grafikę i dużo tekstu do przeczytania. Produkcja została wydana w 1980 r. i wyróżniała się właśnie grafiką, choć dzisiaj pewnie nie będziemy jej oceniać zbyt pozytywnie.

Gra rozpoczyna się w pobliżu opuszczonej posiadłości, w której zostajemy uwięzieni. W budynku znajduje się wiele pomieszczeń i kilka innych osób. Wkrótce wszyscy ludzie giną w niepokojących okolicznościach i okazuje się, że po domu grasuje morderca. Musimy więc odkryć, kim jest morderca, nim sam staniemy się jego następną ofiarą.

Gra została napisana przez Kena Williamsa przez kilka nocy. Gotowa gra nagrana na dyskietkach była początkowo rozprowadzana za pośrednictwem lokalnego sklepu z oprogramowaniem. Sprzedano ponad 10 tys. egzemplarzy, co było wów-



czas niespotykanym fenomenem. W tym latach mało kto sądził, że rynek gier będzie rozwijał się tak dynamicznie. W 1980 r. Ken Williams założył firmę On-Line Systems, która dwa lata później zmieniła nazwę na wszystkim chyba znaną Sierra.

Neuromancer

Teraz skaczemy w czasie, znowu w przyszłość - do 1988 roku. Właśnie

wtedy firma Interplay stworzyła grę, której fabuła została oparta na podstawie powieści Williama Gibsona o tym samym tytule. Akcja gry rozgrywa się w futurystycznej rzeczywistości, a nawet przestrzeni wirtualnej. Przewodnym elementem fabuły jest próba odkrycia prawdy dotyczącej tajemniczych zniknięć grupy hakerów. Pierwszym zadaniem jest odzyskanie komputera, zastawionego w łom-bardzie przez głównego bohatera.





Rozgrywka podzielona jest na trzy tryby. Mamy klasyczną przygodówkę, w której rozmawiamy i handlujemy z innymi postaciami, zbieramy przedmioty i używamy ich na wiele różnych sposobów. Drugi tryb to serwis BBS – przeglądamy fora dyskusyjne, ściągamy pliki i szukamy haseł do innych systemów. Ostatni tryb to cyberprzestrzeń przedstawiona w postaci trójwymiarowej siatki. Poruszamy się pomiędzy systemami komputerowymi, omijamy ich zabezpieczenia, a czasami nawet prowadzimy walki ze sztuczną inteligencją.

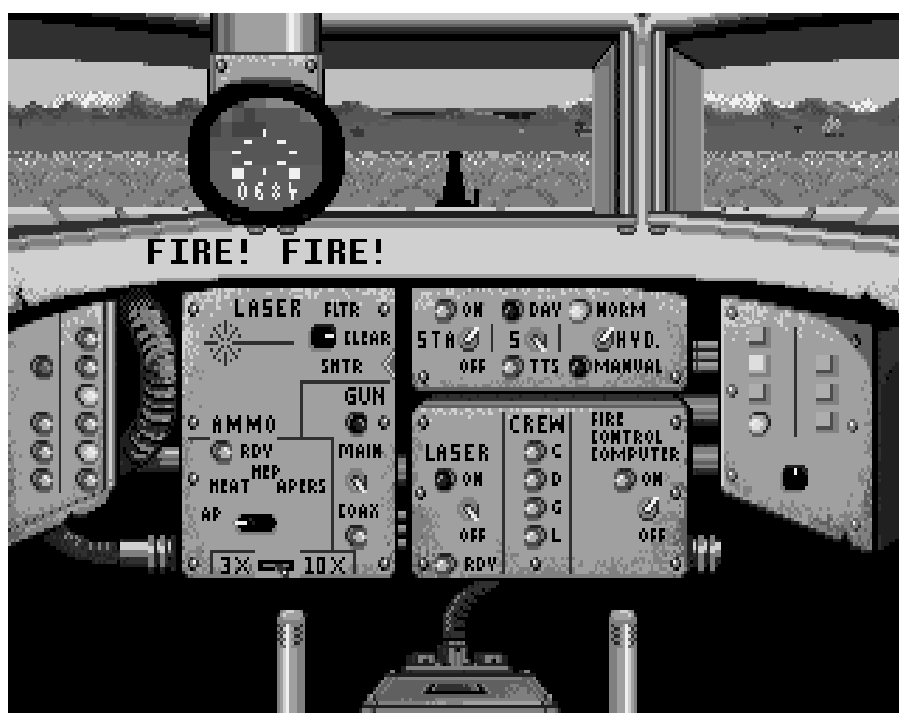
Gra wizualnie prezentuje się dużo bardziej nowocześnie niż wcześniej omówione tytuły. Została wydana także dla innych platform, między innymi Commodore 64 i Amigę. Podczas rozgrywki musimy czytać spore ilości tekstu, dlatego wymaga jest dobra znajomość języka angielskiego. Osobiście bardzo lubię przygodówki i Neuromancer jest jedną z produkcji, do których wracam bardzo często.

Steel Thunder

Czas na inny gatunek elektronicznej rozrywki. Steel Thunder to symulacja wydana w 1988 roku przez słynną

firmę Accolade. Jest to symulator czołgu z perspektywy pierwszej osoby, gdzie gracz bierze udział w fikcyjnych konfliktach zbrojnych. Wcielamy się w dowódcę amerykańskiego czołgu i uczestniczymy misjach na Kubie, w Syrii oraz w Niemczech. Musimy niszczyć różne cele, na przykład bunkry, konwoje,

silosy atomowe, stanowiska artylerii albo większe zgrupowania czołgów. Za pomyślne wykonanie zadań otrzymujemy promocję na kolejne stopnie wojskowe. W każdym państwie do wykonania jest kilka różnych misji, a łącznie do wykonania jest 21 zadań.



W grze mamy do wyboru są cztery istniejące w rzeczywistości czołgi o różnych parametrach i uzbrojeniu: M1A1 Abrams, M3 Bradley, M48A5 Patton oraz M60A3. Przed każdą misją możemy zmieniać parametry takie jak sam czołg, skład załogi, uzbrojenie oraz dodatkowego wyposażenie. Sterowanie czołgiem odbywa się z perspektywy dowódcy, kierowcy lub działonowego. Możemy też powierzyć część zadań członkom załogi kierowanym automatycznie przez komputer.

Ciekawostką jest sterowanie, które jak na datę wydania jest bardzo zaawansowane. Kierujemy czołgiem i jego uzbrojeniem, do którego należy między innymi armata czołgowa, wyrzutnia raketowa i karabiny. Poza tym obsługujemy inny sprzęt, na



więc elementy takie jak tunele, najemnicy czy szpiedzy. Postacie mają różną charakterystykę, a więc określa je siła, inteligencja, szybkość, zwinność, zręczność oraz charyzma. Wszystko razem pozwala na używanie różnych broni oraz trenowaniu różnych umiejętności w celu zdobywania doświadczenia.

W sieci istnieją opinie, że ta produkcja stanowiła inspirację dla twórców serii Fallout. Co ciekawe, w 2014 roku ukazała się druga część gry – Wasteland 2, oczywiście wyłącznie dla peceta. Oczywiście trudno porównywać tu grafikę z pierwowzorem, ale jest to bardzo dobry przykład na to, jak klasyczne produkcje mogą wpływać na obecny rynek gier. Jak wspomniałem we

przykład wyrzutnię granatów dymnych, światła, gaśnice, a także dobiieramy amunicję, kontrolujemy przebieg wydarzeń, wydajemy polecenia podwładnym, sprawdzamy lokalizację na mapie itd.

Steel Thunder to produkcja, która została bardzo dobrze przyjęta przez graczy. Otrzymała bardzo wysokie noty za oprawę graficzną oraz za umiejętne połączenie gry akcji i strategii. Ja zawsze zwracałem uwagę na rozmach i stopień rozbudowy tej gry, co nie było zadaniem łatwym do osiągnięcia, biorąc pod uwagę zasoby sprzętowe komputerów 8-bitowych. Oprócz wersji dla Apple II i Commodore 64, produkcja ta ukazała się również dla systemu MS-DOS.

Wasteland

Ostatnią grą, o której chcę powiedzieć kilka słów to "Wasteland". Została wydana przez firmę Electronic Arts w 1988 roku, także dla Commodore 64. Akcja gry ma miejsce w roku 2087. Po wybuchu trzeciej wojny światowej wygląd Ziemi znacznie się zmienił. Nielicznym ludziom udało się przetrwać w oddalonych od dużych



miast enklawach. Grupy ludzi, którzy przetrwali wojnę, próbują skontaktować się ze sobą w celu odbudowy dawnej świetności rasy ludzkiej. Wcielamy się w militarną grupę, której zadaniem jest zbadanie szeregu dziwnych zdarzeń, które miały miejsce na pustyni.

Gra ma dosyć jednolitą oprawę graficzną, ale jej zaletą jest fabuła i sposób rozgrywki. Mechanika jest oparta na grach fabularnych, mamy

wstępie, 8-bitowe komputery Apple to nie jest sprzęt, dla którego powstały tysiące gier. W polskim internecie znajdziemy listę ok. 40 produkcji, a na angielskich stronach doliczymy się ponad 100 tytułów.

Być może jednak mała ilość przechodzi w jakość? Sprawdźcie to sami, szczególnie jeśli macie do dyspozycji oryginalny sprzęt.

Kamil Stokowski



Kontroler Tandem

Tandem to kontroler napędu CD-ROM przeznaczony dla Amig 600, 1200 oraz 2000, 3000, 4000. Istnieją dwie wersje omawianego urządzenia: jedna dla Amigi 600 oraz 1200 i jedna dla Amig stacjonarnych. Wersja dla Amig 2000, 3000, 4000 pozwala oprócz napędu CD-ROM zainstalować dodatkowy twardy dysk.

W komplecie wraz z Tandemem nie znajdziemy taśmy przeznaczonej do podłączenia napędu oraz (w przypadku Amig 600 i 1200) kabla przeznaczonego do zasilania CD-ROM. W trakcie wpinania taśmy napędu optycznego do gniazda karty musimy uważać, aby nie powyginać niczym nie osłoniętych nóżek gniazd. Zwróćmy także uwagę na prawidłowe ułożenie wtyczki tak, aby żadna nóżka gniazda nie wystawała z boku. Tandem w wersji PCMCIA instalujemy w złączu Amigi 600 i 1200. W przypadku "dużych" Amig mamy do dyspozycji slot Zorro.

Aby używać Tandemu, musimy mieć system operacyjny co najmniej w wersji 2.0 oraz przynajmniej 1 MB wolnej pamięci. Jest to minimalna konfiguracja, zalecany jest system w wersji 3.0 oraz większa ilość pamięci RAM.

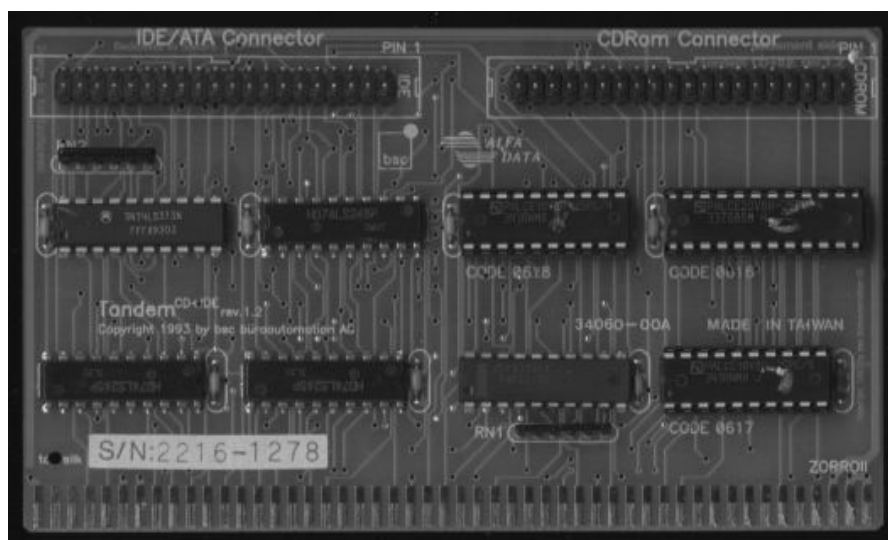
Karta pozwala na wykorzystanie większości napędów CD-ROM. Tandem w wersji dla Amig 2000, 3000 i 4000 pozwala na podłączenie dodatkowego twardego dysku IDE. Należy zwrócić uwagę na fakt, że karta nie pozwala na uruchomienie systemu z tego dysku. Musimy zatem mieć jeszcze drugi dysk w komputerze.

Spowoduje to drobny kłopot w Amidze 2000, nie ma ona bowiem standardowo żadnego kontrolera twardego dysku. Jeśli zatem chcemy startować system z dysku, musimy zakupić dodatkowy kontroler - oprócz Tandemu. Drugą możliwością jest uruchomienie odpowiednio przygotowanej dyskietki startowej, ale nie jest rozwiązanie wygodne. Można je

natomiast traktować jako "tryb awaryjny", gdy chcemy tylko odczytać dane z podłączonego dysku lub płyty CD.

Wraz z kartą otrzymujemy oprogramowanie do obsługi napędów CD-ROM. Dyskietka ta powinna być opisana jako "CD1200Install". Po jej włożeniu do stacji na pulpicie pojawi się ikona "Install-TandemCD". Do obsługi napędów optycznych można także wykorzystać inne oprogramowanie. Tak więc jeśli mamy nasz kontroler niekompletny, możemy sobie poradzić instalując pakiet o nazwie "Atapi PNP".

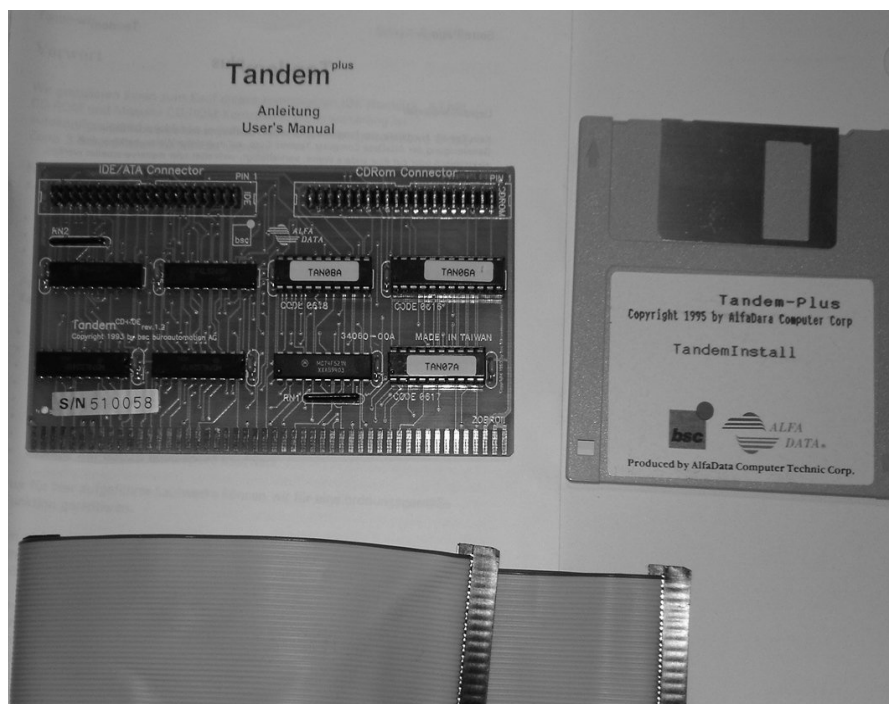
Znajdziemy go oczywiście na Aminecie. Trzeba tylko wyszukać plik o poniższej nazwie:





Atapi_PnP300.1ha

Po rozpakowaniu archiwum należy uruchomić program instalacyjny pakietu. W oknie instalatora wybieramy od razu przycisk "Proceed with install". Pojawi się kolejne okno, w nim wybieramy "Proceed". Teraz pojawi się okno przedstawiające podstawowe informacje o pakiecie. Ponownie wybieramy przycisk z napisem "Proceed".



Teraz trzeba jeszcze dwukrotnie potwierdzić operację za pomocą przycisku "Proceed". Wszystkie potrzebne dane zostaną skopiowane na twardy dysk. W kolejnych dwóch oknach wybieramy znowu "Proceed". W ten sposób spowodujemy zakończenie procesu instalacji.

Zasadniczym elementem pakietu Atapi PNP jest oprogramowanie obsługujące napęd CD-ROM. Powoduje ono, że możemy korzystać z napędu na podobnej zasadzie, jak z twardego dysku. Oprócz tego jednak w skład pakietu wchodzi jeszcze dodatkowe oprogramowanie.

Pakiet zawiera program "CD_IDE", który pozwala zmieniać standardowe

ustawienia napędu optycznego. Standardowo, sterownik CD-ROM sprawdza automatycznie, czy sprzęt jest podłączony. W przypadku niektórych napędów może to być jednak niemożliwe. W razie nieprawidłowej pracy należy wybrać opcję "Disablecheck". W tym wypadku będziemy zmuszeni dodatkowo określić sposób obsługi urządzeń. Przeznaczony jest do tego przycisk cykliczny, na którym na początku

znajduje się napis "Master". Przy użyciu znaczników umieszczonych niżej należy wybrać ustawienie ATA, potem wybrać wspomniany przycisk cykliczny i wskazać na nim ustawienie ATAPI.

Atapi PNP daje możliwość automatycznego zatrzymywania napędu, gdy nie jest on używany. Dzięki temu możemy zmniejszyć zużycie sprzętu. Przeznaczony jest do tego znacznik opisany jako "Idle Stop", który powinien być zaznaczony. Obok znajduje się suwak, który pozwala ustawić, po jakim czasie ma nastąpić zatrzymanie.

W skład oprogramowania pakietu wchodzi także programy "Open",

"Close" oraz "Door". Każdy z nich można uruchomić tak, jak program "CD_IDE". "Open" - powoduje otworzenie szuflady napędu, a "Close" - jej zamknięcie. Program "Door" powinien natomiast działać jak przełącznik, czyli powodować zamknięcie otwartej szuflady oraz otwarcie jej, gdy jest zamknięta. Niestety funkcja ta nie z każdym napędem działa prawidłowo.

Atapi PNP jest wyposażony w program o nazwie "Map", który pozwala odtwarzać płyty dźwiękowe. Możemy go uruchomić podobnie jak wszystkie opisane wcześniej programy.

Ponadto mamy do dyspozycji program o nazwie "PlayCDXL", który służy do odtwarzania filmów w formacie CDXL. Jest to specjalny format filmów stosowany dzisiaj dość rzadko i do tego wyłącznie na Amidze.

Gdy nasz napęd CD-ROM posiada taką możliwość, możemy skopiować ścieżki dźwiękowe na twardy dysk. Przeznaczony jest do tego program o nazwie "TrkDownload",

Do kart Tandem był dołączany pakiet "Cache CDFS", ale może on także obsługiwać dowolny inny kontroler IDE. Musimy jednak zaopatrzyć się w sterownik, czyli plik z rozszerzeniem ".device", który będzie przeznaczony dla kontrolera w naszej Amidze. Sterownik zarządza transmisją danych, a "Cache CDFS" zajmie się rozpoznawaniem formatu płyt i dalszymi operacjami niezbędnymi do pracy.

Po uruchomieniu programu instalacyjnego, pierwszym oraz drugim oknie wybieramy od razu przycisk "Proceed". Uruchomi się program FindCD przeznaczony do ustawiania sterownika napędu.

Na liście oznaczonej jako "CD Device" znajdziemy nazwy sterowników napędów CD-ROM. Jeśli używamy sprzętu podłączanego za pośrednictwem karty Tandem, należy wybrać pozycję:



```
tandemcd.device  
lub  
tandemcd_come.device
```

Oznaką prawidłowego rozpoznania napędu jest komunikat inny niż:

Nothing to select

który widać na liście umieszczonej po prawej stronie okna programu. Gdy nie używamy karty Tandem, musimy skorzystać z innego sterownika.

Po prawidłowym rozpoznaniu napędu CD-ROM, wybieramy przycisk "Use". W kolejnym oknie naciskamy od razu przycisk "Proceed". Kolejne okno będzie zawierało pytanie o to, czy chcemy, aby napęd był uruchamiany wraz ze startem całości systemu. Należy tu wybrać przycisk "Yes", bowiem samodzielne uruchamianie oprogramowania dla napędu jest niewygodne. W następnym oknie zawierającym pytanie o nazwę urządzenia wybieramy przycisk "Proceed". Teraz pojawi się okno przeznaczone do ustawiania katalogu, w którym mają się znaleźć dodatkowe programy. Możemy wybrać od razu "Proceed".

Kolejne okna będą zawierały pytania o chęć instalacji poszczególnych programów składających się na pakiet. W kolejnych dziewięciu (sic!) oknach ponownie wybieramy przycisk "Proceed". Na koniec pojawi się okno z pytaniem, na które odpowiadamy twierdząco - "Yes". Pojawi się jeszcze jedno okno informacyjne, w którym znowu wskazujemy przycisk "Proceed" i instalacja jest zakończona. Do zmiany standardowych ustawień "Cache CDFS" przeznaczony jest program o nazwie "CDFSPrefs". Pozwala on ustawiać ilość pamięci, jaką program obsługujący napęd CD-ROM będzie wykorzystywał do różnych operacji.

Możemy zatem ustawić ilość pamięci przeznaczoną do operacji odczytu danych. Przeznaczona jest do tego ramka opisana jako "Buffers". Musimy najechać na nią wskaźnikiem



myszki i nacisnąć lewy pprzycisk. W ramce pojawi się kursor - trzeba tutaj wprowadzić tam odpowiednią wartość. W zasadzie, im większa liczba, tym szybciej napęd będzie odczytywał dane. Powyżej pewnej wielkości nie zauważymy już jednak przyspieszenia, będzie to spowodowane granicą możliwości kontrolera zainstalowanego w komputerze. Możesz także eksperymentować z poniższymi wartościami:

- Data Cache,
- Line Cache,
- Min Direct.

Liczba Min Direct będzie miała wpływ na szybkość odczytu tylko w sytuacji, gdy dany program będzie odczytywał dane bezpośrednio z napędu. Większość programów robi to jednak za pośrednictwem sterownika. Program "CDFSPrefs" pokazuje także, ile pamięci zajmą nasze ustawienia. Na ilość zajętej pamięci wskazuje pole "RAM kBytes".

Ustawienia dotyczące formatu płyty znajdziemy poniżej napisu "Scan first for". Jeszcze niżej znajduje się przycisk cykliczny. Na początku widoczny jest na nim napis "ISO 9660", a po zmianie będzie to opcja "MAC HFS". Ustawienia te pozwalają określić, jakiego formatu płyty komputer ma

poszukiwać najpierw, gdy płyta zostanie włożona do napędu. Normalnie, ustawienie "Scan first for" nie ma żadnego znaczenia, bowiem oprogramowanie samoczynnie rozpoznaje także kompakty Macintosha. Omawiane funkcje mogą się jednak przydać, gdy będziemy mieli do czynienia z płytą częściowo zapisaną "normalnie", a częściowo w formacie Macintosha.

W programie "CDFSPrefs" znajdziemy także ustawienia dotyczące nazw plików (po prawej stronie). Za ich pomocą możemy decydować o sposobie pokazywania nazw plików i katalogów zapisanych na płytach. Możemy sprawić, aby wszystkie nazwy plików były wypisywane małymi literami. W tym celu wybieramy przycisk opisany jako "Files to Lowercase". Możemy także zdecydować, aby nazwy samych płyt były wypisywane małymi literami, służy do tego znacznik opisany jako "Volumes to Lowercase". Istnieje także możliwość przedstawiania wszystkich nazw z pierwszą wielką lub małą literą, w zależności od ustawienia dwóch poprzednich znaczników. Aby pierwsza litera była mała lub wielka, wybieramy przycisk "Convert First Char".

Adam Zalepa



LHArchie

Jednym z programów ułatwiającym obsługę archiwizerów jest „LHArchie”. Zgodnie z nazwą, za jego pomocą możesz obsłużyć pliki z rozszerzeniem „.lha”. Zwykle trzeba z nich korzystać w oknie „Shell” lub w menadżerach plikowych, ale wtedy wymagana jest odpowiednia konfiguracja. Teraz będziemy mogli zrobić to dużo szybciej i wygodniej.

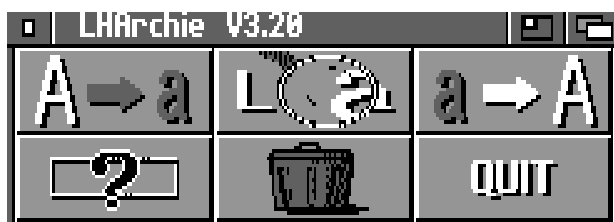
Na początek zajmiemy się instalacją. Z Aminetu pobierz plik o nazwie „LHArchie.lha”, który znajduje się w katalogu „util/arc”. Rozpakuj go, a następnie odczytaj zawartość katalogu docelowego i znajdź kolejny katalog, podpisany tak samo jak nazwa programu. Następnie znajdź program instalacyjny o typowej nazwie „Install”. Przyjmuje on domyślną ścieżkę „SYS:-Tools”, dlatego tam później będziemy szukać głównego pliku programu.

Po jego uruchomieniu na ekranie pojawi się małe okno „Program Error”: w którym przeczytasz stwierdzenie o braku polskiej wersji językowej. W związku z tym użyte zostaną oryginalne angielskie komunikaty. Potwierdź przyjęcie tego do wiadomości przez wskazanie przycisku „Continue”. Otworzy się główne okno programu. Zawiera sześć przycisków, pod którymi ukryte są podstawowe funkcje. Aby uruchomić każdą z nich

należy najechać wskaźnikiem na grafikę i nacisnąć lewy klawisz myszki.

Tworzenie archiwum

Utworzenie nowego archiwum to pierwsza opcja oznaczona przyciskiem zawierającym symbole „A → a”. Po wywołaniu, w górnej części ekranu, pojawi się małe okno z trzema przyciskami. Musisz tu wskazać, czy chcesz poddać kompresji tylko jeden plik czy będzie ich więcej. W tym celu wybierz przycisk „One” (jeden plik) lub „Several” (wiele plików). Możesz także zrezygnować z wykonania operacji przez użycie pola „Cancel”. W przypadku pakowania pojedynczej pozycji z dysku, na ekranie pojawi się zwykłe okno wyboru, z jakim mamy do czynienia przy większości operacji związanych z odczytywaniem lub zapisywaniem danych. Należy w nim wskazać plik, który ma zostać dodany do archiwum i potwierdzić wykonanie za pomocą przycisku „OK” umieszczonego niżej.



Następnie zobaczysz identyczne okno, lecz zatytułowane „Enter archive name”. Ustaw tu katalog, w którym zostanie zapisane archiwum oraz wprowadź nazwę nowego pliku. Domyślnie podana będzie już nazwa pliku źródłowego uzupełniona o rozszerzenie „.lha”, wystarczy więc

ją zachować lub zmodyfikować na bardziej odpowiednią. Plik może być zapisany pod dowolną nazwą, jedynym ograniczeniem są możliwości zastosowanego systemu plikowego. Dalej jeszcze raz użyj przycisku „OK”. Program rozpocznie tworzenie archiwum, lecz w trakcie nie będą wyświetlane żadne komunikaty. Musisz poczekać na zakończenie pracy, co może potrwać kilka sekund lub nawet wiele minut, bowiem zależy to od rozmiaru pliku i szybkości procesora w Twojej Amidze.

Gdy operacja zostanie zakończona na ekranie pojawi się okno „LHArchieConsole” zawierające szczegółowe informacje o archiwum. Są to te same komunikaty, które w zwykły sposób widoczne są w oknie „Shell”. Gdy się pojawią mamy pewność, że plik został prawidłowo zapisany na dysku.

Sytuacja wygląda inaczej jeśli kompresujesz większą ilość plików, czyli użyjesz opcji „Several”. Tym razem zobaczysz większe okno o nazwie „File selection window”, wewnątrz dużą listę, a pod nią kilka przycisków. Służy ona do dodawania pozycji z katalogu, które znajdą się później w archiwum. W tym celu skorzystaj z funkcji „Add”, a potem za pomocą kolejnego okna wyboru wskaż interesujące Cię pliki.

Analogicznie możesz dodać wszystkie pozycje z konkretnego katalogu, lecz należy użyć przycisku „Add dir”. Program automatycznie odczyta nazwy plików zapisanych w katalogu. Czynności te powtórz tyle razy ile



potrzebujesz, aby na liście znalazły się wszystkie pliki. Będzie ona zawierać ich nazwy oraz ścieżki dostępu.

Jeżeli stwierdzisz, że chcesz skasować niektóre pozycje z listy, najedź wskaźnikiem na nazwę przeznaczoną do usunięcia i naciśnij lewy klawisz myszki. Powinna zostać podświetlona tak jak na ilustracji. Następnie użyj przycisk „Remove” na dole. Zwróć uwagę, że małe pole w górnej części opisane jako „chosen files” pokazuje ilość wybranych plików. Zawartość listy możesz przewijać za pomocą suwaka umieszczonego po prawej stronie. Gdy ustalisz całą zawartość archiwum, wybierz przycisk „Ok”.

Dalsze zachowanie programu jest identyczne jak podczas dodawania do archiwum jednego pliku. Pojawi się okno wyboru, w którym wskaż katalog i nazwę pliku archiwum, po czym użyj przycisku „OK”. Rozpocznie się właściwa operacja kompresji danych. Pamiętaj, że jej czas w głównej mierze jest uzależniony od łącznej objętości wskazanych plików, a nie od ich ilości. Dlatego cierpliwie poczekaj na zakończenie pracy. Potwierdzi to okno z podsumowaniem „LHArchieConsole” - identyczne jak przy pakowaniu pojedynczych pozycji.

Wyświetlanie informacji o archiwum

Aby zobaczyć jakie pliki zostały umieszczone w archiwum trzeba skorzystać z przycisku przedstawiającego „lupkę” w głównym oknie programu. Na ekranie znowu pojawi się okno wyboru, w którym wskazujemy plik z rozszerzeniem „.lha”. Zwróć uwagę, że w polu filtra AmigaDOS utworzony jest wpis: „#?.lha”, co oznacza, że okno będzie pokazywać wyłącznie te pliki, które są możliwe do odczytania przez „LHArchie”. Jest to ułatwienie dla początkujących użytkowników. Jeżeli chcesz zobaczyć jakie inne pozycje znajdują się na dysku, skasuj całą

treść pola „Filtr” lub pozostaw tylko symbol „#?”.

Dalej użyj przycisku „OK” w dolnej części okna. Po chwili pojawi się okno informacyjne, w którym wyświetlona będzie zawartość archiwum. Gdy zapoznasz się z nimi po prostu zamknij okno. Zwróć uwagę, że jeśli plik „.lha” zawiera większą ilość danych, na ekranie mogą się nie zmieścić wszystkie informacje. Przewijanie komunikatów możesz jednak zatrzymać naciskając SPACJĘ, a ponownie kontynuować klawiszem BACKSPACE. Jest to uniwersalny sposób, o którym mówiliśmy też w punkcie omawiającym sam archiwizator „LHA”.

Rozpakowanie archiwum

Funkcja rozpakowania plików jest dostępna przy użyciu przycisku zawierającego symbole „a → A”. Na ekranie pojawi się okno wyboru, w którym należy wskazać plik przeznaczony do dekompresji, po czym potwierdzić operację przyciskiem „OK”. Następnie zobaczysz okno „File selection window”, bardzo podobne do tego, które wyświetlane

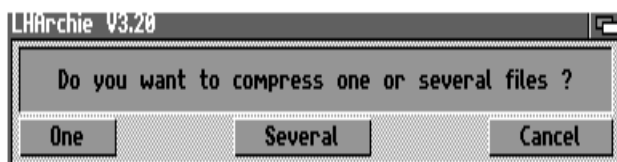
Zdecyduj więc które pozycje chcesz rozpakować, a potem wybierz przycisk „Ok”. Dalej program poprosi jeszcze o wskazanie katalogu docelowego przez wyświetlenie znajomego okna wyboru. Wykonana zostanie dekompresja, po której zobaczysz duże okno informacyjne zawierające szczegóły związane z rezultatami czynności. Zwróć uwagę, że znowu są to te same dane, które normalnie uzyskujemy w oknie AmigaDOS.

Należy także zwrócić uwagę na pewną niedoskonałość programu. Funkcje wyświetlania zawartości oraz rozpakowania archiwum nie zawsze wypisują prawidłowo listę plików w oknie „File selection window”. Ma to związek m.in. z wersją polecenia „LHA”, bowiem musi ono znajdować się w systemowym katalogu „C”, aby „LHArchie” działał prawidłowo. W przeciwnym razie wybór przycisków w głównym oknie zakończy się brakiem reakcji lub wywołaniem błędu systemowego. Podobne kłopoty występują niekiedy w sytuacji, gdy ilość dostępnej pamięci jest zbyt mała na umieszczenie wszystkich pozycji na liście.

Na szczęście nie ma to wpływu na pozostałe opcje i niezależnie od tego, czy lista plików będzie widoczna – zarówno kompresja, jak i dekompresja przebiegnie prawidłowo. Polecam korzystać z archiwizera „LHA” dostępnego na Aminecie. Autor programu zaleca dwie wersje – 1.38 oraz 1.54.

Usuwanie plików z archiwum

Podobne problemy mogą wystąpić podczas wywoływania kolejnej funkcji programu modyfikującej gotowy plik „.lha”. Pozwala ona skasować niepotrzebne pliki z archiwum bez potrzeby dekompresji. W tym celu wybierz przycisk w głównym oknie zawierający grafikę Śmietnika.



jest podczas pakowania danych. Będzie teraz zatytułowane „Files to extract”, a aktywne będą trzy funkcje umieszczone niżej: „Ok”, „Remove” oraz „Cancel”. Za pomocą pierwszej potwierdzamy chęć rozpakowania archiwum w całości.

Drugi przycisk pozwala usunąć określone pozycje z listy, dzięki czemu zostaną one pominięte i nie będą zapisywane na dysku. Aby tak się stało najpierw podświetl wybrane pliki w taki sam sposób jak wcześniej. Ostatnia opcja oznacza zrezygnowanie z wykonania operacji w ogóle.



Na ekranie pojawi się okno wyboru, w którym wskaż plik archiwum, a następnie skorzystaj z przycisku „OK”. Po chwili zobaczysz znajome okno „File selection window”, lecz lista zatytułowana będzie „Files to delete”. Pozostała część funkcji jest analogiczna do opcji rozpakowania danych i ich obsługa jest taka sama.

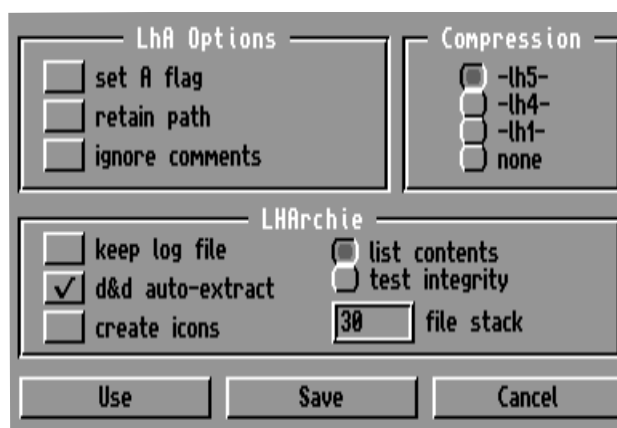
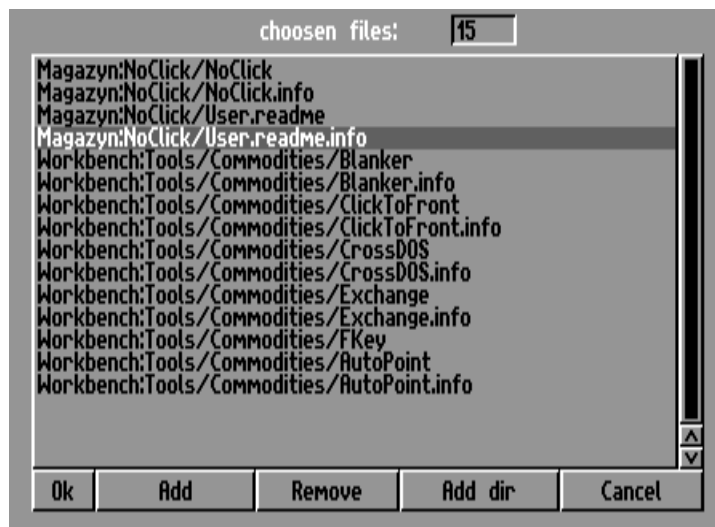
Pamiętaj tylko, że podświetlone pozycje zostaną bezpowrotnie usunięte z archiwum i nie będzie możliwości ich odzyskania. Dlatego operacje te wykonuj z dużą ostrożnością.

Ustawienia programu

W głównym oknie dostępny jest także przycisk zawierający grafikę podobną do standardowej ikony systemowego katalogu „Prefs”. Przedstawiamy go na ilustracji obok. Za jego pomocą możemy zmienić konfigurację niektórych funkcji „LHArchie”. Najedź na niego wskaźnikiem myszki i naciśnij lewy klawisz. Zobaczysz okno ustawień zawierające następujące opcje.

Zwróć uwagę na grupę opisaną jako „Compression”. Możesz w niej zmieniać metodę pakowania danych lub zastosować archiwizację bez kompresji (pozycja „none”). Omówiliśmy je szczegółowo w części poświęconej samemu programowi „LHA”. Za pomocą pola „set A flag” możesz spowodować, że pliki dodane do archiwum będą oznaczone przez „włączenie” atrybutu A.

Dzięki temu będziesz mógł łatwo sprawdzić, które pozycje na dysku zostały spakowane. Opcja o nazwie „retain path” powoduje, że podczas wykonywania wszelkich operacji program będzie zawsze zachowywał



pełne ścieżki dostępu zamiast tylko nazw plików. Pozwala to odtworzyć w ramach archiwum całą strukturę katalogu, co jest często niezbędne, na przykład podczas kompresji katalogu z bardziej rozbudowanym programem.

Warto również przyrzeć się funkcji oznaczonej jako „create icons”. Jeśli będzie aktywna, tworzone archiwa będą uzupełniane o odpowiadające im ikony. Daje to korzyść w tej postaci, że na Workbenchu możesz je uzupełnić o wpisy pozwalające rozpakować dane za pomocą zwykłego „dwukliku”. Aby było to możliwe musisz wywołać okno informacyjne, a następnie w odpowiedni sposób uzupełnić parametry ikony.

Dodajmy, że ikona powinna posiadać rodzaj „Projekt” (ang. „Project”) i zawierać odniesienie do programu

„LHArchie” albo innego podobnego, dysponującego interfejsem graficznym. Na przykład, jeśli w polu „Program” (ang. „Default Tool”) podasz nazwę programu wraz ze ścieżką dostępu:

SYS:Tools/LHArchie/LHArchie

po „dwukliku” na ikonie zostanie automatycznie wyświetlone okno wyboru katalogu docelowego. Gdy go ustawisz i wskażesz przycisk „OK”, program rozpakuje archiwum.

Ścieżka dostępu:

SYS:Tools/LHArchie

jest oczywiście przykładowa, musisz tu wpisać katalog, w którym zainstalowałeś wcześniej program. Funkcja ta jest więc sporym ułatwieniem dla osób mniej zaznajomionych z systemem Amigi.

Ostatni przycisk w głównym oknie programu zawiera napis „QUIT” i służy do zakończenia pracy. Wystarczy najechać na niego wskaźnikiem i nacisnąć lewy klawisz myszki, a „LHArchie” zostanie wyłączony.

Zwróć uwagę, że nie będzie wyświetlone żadne pytanie o potwierdzenie. Alternatywnie możesz po prostu zamknąć okno przy użyciu standardowego przycisku w lewym górnym rogu.

Program jest wygodnym narzędziem, szczególnie dla osób dysponujących mniej rozbudowaną konfiguracją sprzętową Amigi. Nie jest bardzo rozbudowany, ale jednocześnie zajmuje małą ilość pamięci i nie wymaga instalacji dodatkowych składników w systemie.

Adam Zalepa



SCALOS

Lepszy blat

Scalos jest zastępczym pulpitem, czyli zamiennikiem Workbencha. Posiada własny menedżer okien, ponadto naśladuje funkcje Workbencha, ale posiada więcej możliwości i ulepszony wygląd interfejsu użytkownika. Scalos był kiedyś komercyjnym produktem, został napisany w 1999 roku przez programistę Stefana Sommerfiel-da. Jego celem było stworzenie alternatywy dla interfejsu Workbencha, który już wtedy był uznawany za zbyt mało rozbudowany dla potrzeb bardziej wymagających użytkowników.

Instalacja Scalosa jest bardzo łatwa, wystarczy skorzystać z dołączonego programu instalacyjnego. Kiedy instalacja systemu ScalOS dobiegnie końca, możemy po raz pierwszy uruchomić Scalos. W tym celu w oknie "Wykonaj polecenie" należy wpisać:

SCALOS:Scalos

Wraz ze Scalosem na dysku instalowany jest szereg dodatkowego oprogramowania. Większość to unowocześnione wersje programów przeznaczonych do kontroli ustawień,

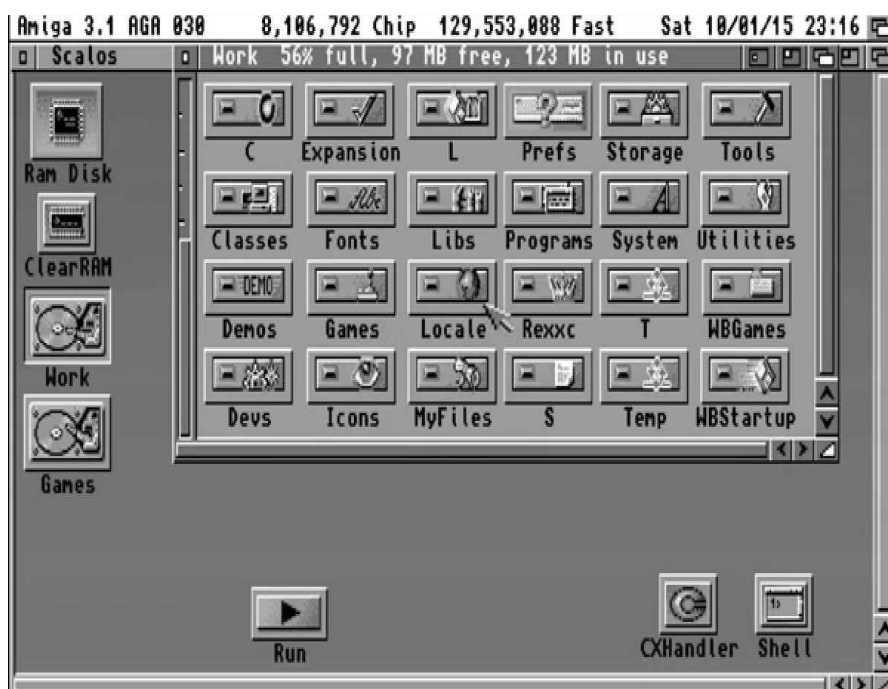
które znajdują się także na wyposażeniu Workbencha. Oczywiście, uwzględniają one funkcje, które posiada Scalos, a standardowo nie posiada Workbench.

Zaraz po pierwszym uruchomieniu Scalosa, żadne ustawienia nie są wybrane, dlatego system wygląda dość ubogo. Można mieć wtedy wrażenie, że praca będzie bardzo niewygodna. Jednak Scalos ma naprawdę duże możliwości, trzeba go tylko odpowiednio skonfigurować.

Na początek powinniśmy wybrać ustawienia menu górnego. Bez nich

poruszanie się po systemie jest bardzo utrudnione. Scalos na początku nie ma menu w ogóle. Dzięki temu możemy mieć całe menu pod całkowitą kontrolą. Cecha ta z pewnością wyróżnia Scalos spośród wszystkich programów pozwalających uzyskać graficzne środowisko użytkownika na Amidze.

Musimy uruchomić program o nazwie "Scalos Menu", który pozwoli wybierać ustawienia. Znajduje się on na dysku systemowy - tam, gdzie była wykonywana instalacja pakietu. Możemy utworzyć własne pozycje menu lub szybciej - zainstalować go-





to ustawienia. Menu może także uruchamiać wewnętrzne funkcje Scalosa. W większości przypominają one opcje Workbench'a lub Directory Opusa, tak więc ich obsługa nie powinna nastręczać trudności.

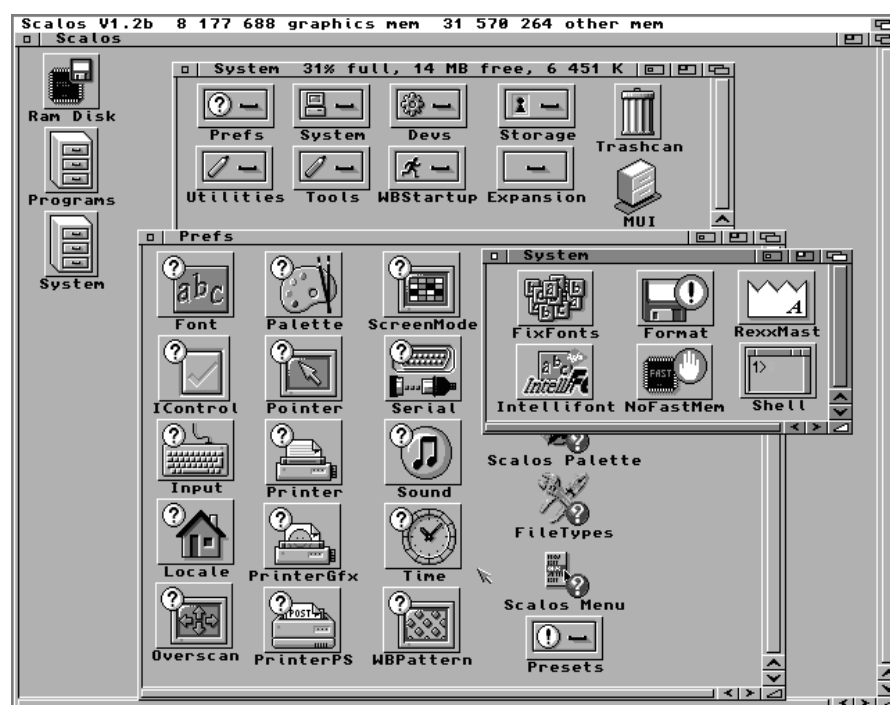
Gdy już utworzymy odpowiednie menu, możemy zmieniać położenie jego opcji. Aby to zrobić wystarczy je przesunąć, podobnie jak ikony na pulpicie. W identyczny sposób możliwe jest także przemieszczanie całego menu. Zamiast tytułu opcji trzeba "złapać" tytuł menu.

Scalos pozwala ustawiać kolory swojego pulpitu, podobnie jak Workbench. Przeznaczony jest do tego kolejny program - o nazwie "Scalos Palette". Umożliwia on dostęp do poniższych funkcji:

- ustawianie dowolnej ilości kolorów na pulpicie tak, by miały określone, konkretne wartości,

- przypisywanie kolorów różnym elementom interfejsu użytkownika, na przykład koloru ramki aktywnego okna, ważnego tekstu, zwykłego tekstu itp.

Pakiet pozwala również zmieniać napisy umieszczane standardowo na



listwach tytułowych okien oraz ekranu.

Możemy zmienić:

- napis znajdujący się na listwie tytułowej ekranu, na którym znajduje się pulpit,
- napisy na listwach tytułowych okien katalogów dysków,
- napisy na listwach tytułowych pozostałych okien wyświetlających zawartość dysków.

Warto zwrócić uwagę, że w każdym wypadku napisy mogą być inne, a do ich tworzenia możemy użyć wiele znaków specjalnych, które mają podobne funkcje jak w Directory Opusie. Scalos jest całkowicie zgodny z AmigaOS. Zachowuje się identycznie jak Workbench, stanowiąc jednocześnie jego unowocześnioną wersję. Wykorzystuje nowe elementy systemu, takie jak pakiet "Magic User Interface". Zachowuje bardziej nowoczesny sposób wybierania ustawień. Pakietu można używać zamiast oryginalnego blatu, w tym celu zamiast korzystać z polecenia LOADWB należy uruchomić Scalos.

Zamienniki Workbench'a to temat rzeka. Piszę o nich od wielu lat, ale wcześniej zwykle zwracałem uwagę na bardziej popularnego Directory Opusa. Scalos należy do programów nietypowych, bowiem jego autor skupił się bardziej na funkcjach pulpitu, niż typowego menedżera plików. Gdy mamy bardziej rozbudowaną konfigurację Amigi albo korzystamy z emulatora, możemy sprawić, że nasz pulpit będzie dużo bardziej atrakcyjny i funkcjonalny niż Workbench.

Adam Zalepa





New Mode

Gdy zainstalujesz kartę graficzną szybko przekonasz się, że nie wszystkie programy mają możliwość zmiany trybu wyświetlania. Niektóre z nich uparcie nie akceptują nowej konfiguracji i wciąż korzystają ze starych układów graficznych na płycie głównej Amigi. Możesz to zmienić przez użycie tzw. „promotorów wyświetlania”, czyli narzędzi zmieniających parametry ekranu, na którym uruchamiają się inne programy.

Funkcja ta działa poprawnie w stosunku do większości pozycji korzystających z systemu operacyjnego. Niestety można też spotkać takie, których nie zmusimy do zmiany trybu wyświetlania, ale są to w większości starsze gry. Podczas używania oprogramowania użytkowego problem ten występuje rzadko.

Najprostszy „promotor” jest wbudowany w fabryczny program „IControl” znajdujący się w katalogu „Prefs” na dysku systemowym. Mamy tu opcję „Promocja trybów” (ang. „Mode Promotion”), której aktywacja ma powodować, iż tryby „PAL” i „NTSC” będą zastępowane przez bliźniacze „DbIPAL” i „DbINTSC”. Pliki te muszą znajdować się w katalogu „Monitors” umieszczonym w systemowym urządzeniu „DEVS:”. Funkcja wymaga układów „AGA”, które spotkamy w Amidze 1200 i 4000, ponadto nie działa w przypadku pozostałych trybów wyświetlania. Jest więc możliwa

do zastosowania tylko w niektórych przypadkach.

Dlatego proponuję skorzystać z lepszego „promotora wyświetlania” o nazwie „NewMode”. Jest on w pełni zgodny w systemem operacyjnym, nie powoduje komplikacji w używaniu oprogramowania, nie zajmuje też wiele pamięci. Jest wygodny i radzi sobie bardzo dobrze z ustawianiem nowych trybów wyświetlania, według życzeń użytkownika.

Program znajduje się na Aminecie w katalogu „util/cdity” pod nazwą „NewMode_V39.lha”. Pobierz ten plik i rozpakuj w zwykły sposób. Uruchom program instalacyjny i przeprowadź typową instalację programu. W trakcie nie spotkasz żadnych niespodzianek. Następnie zresetuj Amigę i wczytaj ponownie Workbench. Na ekranie, w lewym górnym rogu, powinno pojawić się małe okno

z komunikatem „Nie mogę wczytać preferencji”. Jest to normalne, bowiem nie zapisaliśmy jeszcze konfiguracji nowego programu. Aby to zrobić musisz wywołać okno ustawień.

„NewMode” należy do kategorii oprogramowania typu „commodity”, dlatego można go kontrolować za pomocą ich systemowego „menadżera”. Dlatego w następnym kroku uruchom program „Exchange”. Na ekranie pojawi się okno zatytułowane w taki sam sposób. Najedź wskaźnikiem na nazwę „NewMode” widoczną na liście po lewej stronie i naciśnij lewy klawisz myszki. Wybrana pozycja powinna zostać podświetlona, a w prawej części okna zobaczysz podstawowe informacje o programie. Teraz wystarczy już tylko wybrać przycisk „Pokaż interfejs”. Na ekranie zostanie wyświetlone główne okno „NewMode”.





Podstawowa konfiguracja nie jest rozbudowana, gdyż większość funkcji ustawiamy podczas uruchamiania innych programów, korzystających z oddzielnych ekranów. Dopiero wtedy aktywna stanie się lista „Obiekty”. Na razie trzeba tylko zdecydować, czy program ma umożliwiać zmianę parametrów ekranu. W tym celu powinno być aktywne pole „włączona” widoczne w grupie „Promocja” po prawej stronie.

Jeśli chcesz, aby sprawdzane były wszystkie nowo wywoływane ekrany systemowe, „włącz” dodatkowo pozycję „wszystkie” poniżej. Wskazanie pola „nowe” - tak jak na ilustracji – spowoduje, że „NewMode” będzie brał pod uwagę tylko te ekrany, których jeszcze nie zna, czyli w praktyce będzie to dotyczyło nowo uruchamianych programów. Możesz

Gdy ustawisz wszystkie opcje, użyj przycisku „Zapisz” w dolnej części okna. Aby sprawdzić czy wszystko jest w porządku najlepiej jeszcze raz zresetować komputer. Po załadowaniu systemu na ekranie nie powinno pojawić się żadne nowe okno związane z naszym „promotorem”. Jeżeli tak jest możesz przejść do dalszej konfiguracji, tym razem już konkretnych programów.

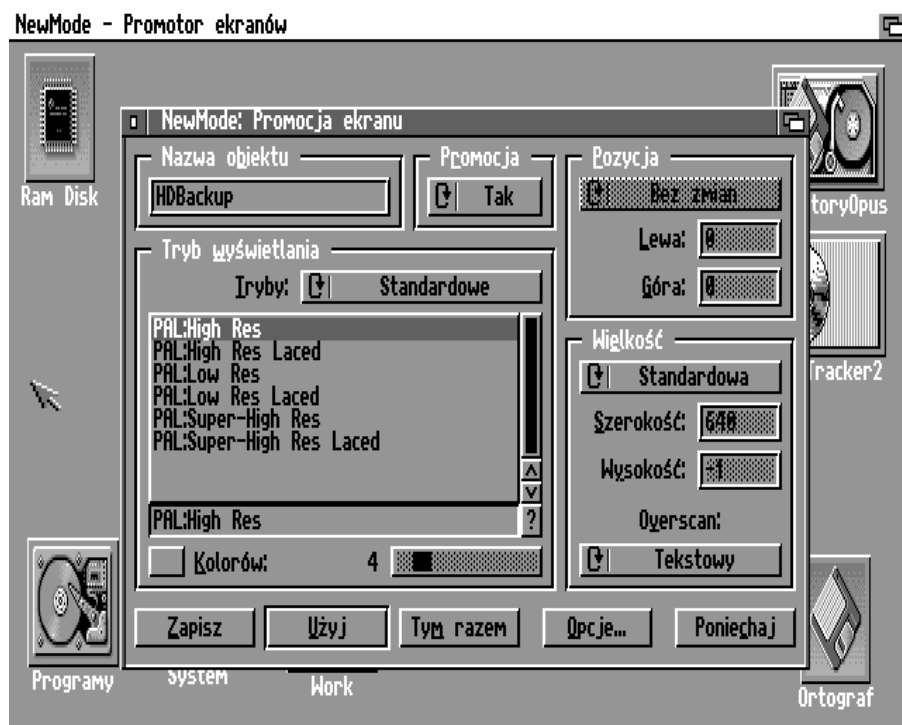
Wbrew pozorom jest to bardzo łatwe i nie wymaga wykonywania wielu skomplikowanych czynności. Należy po prostu uruchomić program, który wywołuje własny ekran. Może być to na przykład systemowy „HDBackup”, który znajduje się w katalogu „Tools”. Po wykonaniu „dwukliku” na jego ikonie, nie pojawiają się jednak opcje programu, lecz nowe okno związane z „NewMode”.

uzależniona od parametrów ekranu, jaki wywołuje uruchamiany program. Za każdym razem możesz je jednak zmienić według własnych potrzeb. W tym celu najedź wskaźnikiem na nazwę trybu i naciśnij lewy klawisz myszki.

Za pomocą przycisku cyklicznego „Tryby” możesz zmieniać ich rodzaje. Najbardziej interesujące są dwie pozycje: „Extra Half Brite” i „Hold And Modify”. Pozwalają one korzystać trybów graficznych Amigi o takich samych nazwach, które normalnie nie są dostępne na Workbenchu. Na liście pojawiają się wtedy dodatkowe oznaczenia „EHB” oraz „HAM”. Więcej o wykorzystaniu tego ostatniego przeczytasz także w części zatytułowanej „Tryb wyświetlania Hold-And-Modify”.

Zwróć uwagę na grupy opisane jako „Wielkość” oraz „Pozycja”. Za pomocą tej pierwszej możliwe jest korzystanie z trybu „Overscan” rozszerzającego obszar roboczy. Dzięki opcjom widocznym w grupie „Pozycja” możesz natomiast – zgodnie z nazwą - zmieniać pozycję ekranu licząc od lewego górnego rogu. Dlatego pola mają nazwy „Lewa” i „Góra”. Wartości zerowe wyznaczają standardowe położenie ekranu, przy czym można stosować liczby zarówno dodatnie, jak i ujemne. Funkcje te przydają się w sytuacji, gdy chcesz skorzystać z nietypowego trybu wyświetlania i część zawartości jest niewidoczna. Możesz ją ustawiać bardzo elastycznie przesuwając uruchamiany ekran systemowy w każdą stronę.

Jeśli „włączysz” pole opisane jako „Kolorów” znajdujące się poniżej listy trybów, aktywny stanie się także suwak obok. Za jego pomocą możesz regulować ilość dostępnych barw, podobnie jak w systemowym programie „ScreenMode”. Okno zawiera jeszcze przycisk „Opcje...”, dzięki któremu uzyskujemy dostęp do możliwości zmiany sposobu rozpoznawania uruchamianego programu. Na ekranie pojawiają się nowe opcje.



także zrezygnować z możliwości konfiguracji na bieżąco. W tym celu użyj pola „żadne” lub „dodawaj do listy”. W tym drugim przypadku każdy nowy ekran zostanie umieszczony na liście, a zmianę jego parametrów można wtedy uzyskać za pomocą przycisku „Edytuj...” widocznego pod listą „Obiekty”.

Funkcje będą takie same podczas uruchamiania każdego programu, który otwiera oddzielny ekran. Różnić się będą tylko informacjami w polu „Nazwa obiektu”, bowiem umieszczana jest tutaj nazwa programu. Opcje w grupach „Tryb wyświetlania” oraz „Wielkość” mogą być inaczej ustawione, gdyż ich konfiguracja jest

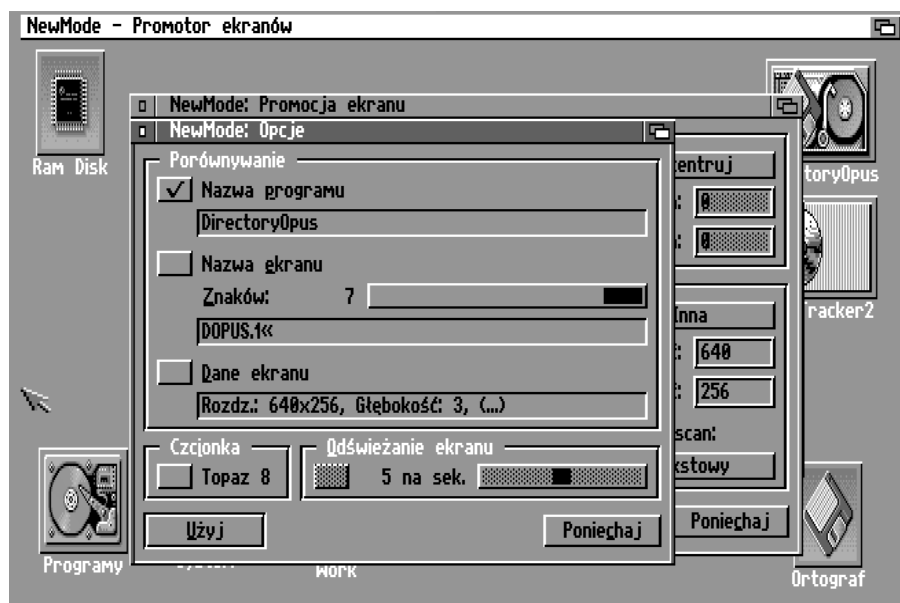


Standardowo „włączona” jest pozycja oznaczona jako „Nazwa programu”, co powoduje, że „NewMode” porównuje kolejno uruchamiane pozycje biorąc pod uwagę nazwę. Możesz również uaktywnić funkcje „Nazwa ekranu” lub „Dane ekranu”. Pierwsza posiada suwak, który pozwala ustawić ilość znaków nazwy ekranu systemowego, które będą porównywane.

Zwróć uwagę, że nazwa programu i nazwa ekranu mogą być zupełnie różne lub podobne, szczególnie przy uruchamianiu kilku kopii tego samego programu jednocześnie. Funkcja „Dane ekranu” powoduje natomiast, że przy porównywaniu brane pod uwagę są parametry wywoływanego ekranu takie jak rozdzielczość czy ilość kolorów, czyli dotyczy to charakterystycznych cech poszczególnych trybów wyświetlania.

Każdą opcję możesz ustawić dodatkowo lub zamiast poprzedniej. Wystarczy tylko „włączyć” lub „wyłączyć” jedno z omawianych pól – działają one niezależnie od siebie. Dzięki nim „promotor” może działać dużo bardziej precyzyjnie albo stosować inne kryteria dla różnych programów.

Na dole okna dostępne jest jeszcze pole „Topaz 8”. Jeśli zostanie zaznaczone, program zostanie uruchomiony w taki sposób, że będzie używał systemowej czcionki „Topaz” w rozmiarze 8. Zostanie to ustawione niezależnie od krojów wybranych na Workbenchu lub wskazanych z innych ustawieniach.



Jest to przydatne w sytuacji, gdy chcesz uruchomić starszą pozycję, która nieprawidłowo obsługuje inne rodzaje czcionek.

Normalnie należałoby zmienić kroje w systemowym programie „Font” (z katalogu „Prefs”) i ponownie uruchomić program sprawiający kłopoty. „NewMode” ułatwia i przyspiesza pracę w tym zakresie.

Gdy zdecydujesz o wszystkich opcjach skorzystaj z jednego z przycisków umieszczonych na samym dole okna. Są one podobne do tych, które znajdziesz w programach z systemowego katalogu „Prefs”.

Po wskazaniu opcji program zostanie uruchomiony przy uwzględnieniu nowych parametrów ekranu. W ten sposób możesz je zupełnie dowolnie zmieniać.

W większości przypadków pozwoli to na wywołanie trybu wyświetlania odpowiedniego dla Twojego monitora lub karty graficznej. Pamiętaj jednak, że rozwiązanie to nie sprawdza się w każdym przypadku. Jeżeli program napisany jest z pominięciem funkcji systemu operacyjnego lub jego działanie będzie ściśle związane z określonym trybem, może okazać się, że nastąpi awaria systemu albo ekran zostanie otwarty w niepoprawny sposób. Niektóre elementy mogą być wyświetlane nieprawidłowo lub nie będą pokazywane w ogóle.

Podobnie niekorzystnie może odbić się zwiększona lub zmniejszona ilość kolorów. Może być bowiem przyjmowana stała paleta przez co, po uruchomieniu na ekranie zawierającym inną ilość kolorów, obraz będzie wyglądał nieprawidłowo. W takich sytuacjach należy spróbować wyłączyć program lub zresetować komputer. Następnie – jeśli zapisałeś ustawienia – musisz zmienić konfigurację lub wyłączyć działanie „promotora” w stosunku do uruchamianej pozycji. W przeciwnym razie będzie działała nieprawidłowo za każdym razem. Aby to poprawić trzeba wywołać ustawienia „NewMode” w sposób, który podaliśmy wcześniej – na przykład za pomocą programu „Exchange”.





Następnie na liście po lewej stronie podświetl pozycję z nazwą problematycznego programu i użyj przycisku „Edytuj...” poniżej. Powrócisz do okna „Promocja ekranu”. Przetwórz nim przycisk cykliczny opisany jako „Promocja” na pozycję „Nie”.

Na koniec zapisz nowe ustawienia. Dzięki temu „NewMode” nie będzie brał pod uwagę możliwości zmiany parametrów ekranu związanego ze wskazanym programem.

Alternatywnie po wywołaniu okna „NewMode” możesz usunąć całkowicie pozycję z listy. W tym celu, zamiast „Edytuj...” skorzystaj z przycisku „Usuń”. Jest to jednak rozwiązanie niezbyt wygodne, bo za każdym razem, gdy uruchomisz program pojawi się nowe okno „promotora”. Będziesz musiał w nim użyć przycisku „Poniechaj”, aby pominąć ustawianie konfiguracji ekranu.

Dlatego najlepiej zachować wszystkie ustalone pozycje na liście, a tylko wyłączyć funkcję zmiany trybów wyświetlania.

Omówiony program jest typowym, lecz oczywiście nie jedynym przedstawicielem tzw. „promotorów wyświetlania”. Jest wygodny, posiada spore możliwości i nie ma dużych wymagań sprzętowych. Podczas konfiguracji mogą pojawić się problemy, ale wynikają one w większości z funkcji uruchamianych programów, a nie niedoskonałości naszego „promotora”. Zmienia on działanie programów w sposób wymuszony, dlatego nie sprawdzi się w każdym przypadku.

Zawsze dużo lepiej ustawić parametry ekranu w konfiguracji konkretnego programu, dlatego z „NewMode” korzystaj tylko wtedy, gdy inne sposoby zawiodły. Można go stosować niezależnie od tego czy posiadasz dodatkową kartę graficzną, czy używasz standardowych układów Amigi.

Adam Zalepa

RETROLAB.PL

POŁĄCZYŁA NAS PASJA DO RETRO

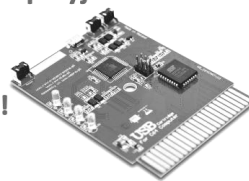
WYKONAMY WSZYSTKO DLA WASZYCH RETRO-MASZYN!



- naprawy, w tym wymiany kondensatorów, poprawki po poprzednich naprawach
- modyfikacje (również w sposób, który sam zaproponujesz)
- odświeżymy plastiki, żebyś mógł dumnie zaprezentować swój retro sprzęt przed przyjaciółmi



- projektujemy peryferia dla Waszych komputerów!



- Organizujemy Retro spotkania i zajęcia dla najmłodszych



- Kupiłeś retro sprzęt z drugiej ręki od "super sprzedawcy" - który twierdzi "nie mam możliwości sprawdzenia dlatego sprzedaje jako uszkodzone".

Uszkodzone? Nie ma problemu my to naprawimy i jeszcze damy gwarancję!

- CHCESZ ZOBACZYĆ JAK PRACUJEMY ZAPRASZAMY NA NASZ FACEBOOK ORAZ STRONĘ WWW



- KONTAKT TEL. 531190100 EMAIL: HELPME@RETROLAB.PL
- MIASTA: BIELSKO-BIAŁA, KATOWICE, GDYNIA

ARHNEU

SAVES
FLIPPERS
www.savetheflippers.com

RETROKLINIK

RETROTRONICS

PIXEL

PATRYCJA KOZIATEK
FOTOGRAFIA



Komunikacja między procesami

System operacyjny Amigi stwarza szerokie możliwości komunikacji między procesami oraz zadaniami. W przypadku procesów, mamy do dyspozycji tzw. "potoki" (ang. pipe) oraz sygnały.

Sygnał zatrzymania

Każdy proces może zostać zatrzymany przez wysłanie do niego sygnału, który możemy wysłać za pomocą naciśnięcia kombinacji klawiszy albo poprzez polecenie AmigaDOS zwanego BREAK.

Zatrzymanie "ręczne"

Gdy uruchomimy jakiegokolwiek polecenie DOS lub program, zwłaszcza w środowisku Shell, można go zatrzymać naciskając kombinację klawiszy CONTROL + C. Spowoduje to wysłanie sygnału zatrzymania i zakończenie tego procesu.

Aby zatrzymać proces przy użyciu polecenia BREAK, musimy skorzystać z poniższego wzoru:

BREAK <numer> C

Gdzie "<numer>" to oczywiście numer procesu, który ma być zatrzy-

many. Skąd jednak mamy wiedzieć, jaki numer ma interesujący nas proces? Można to odczytać za pomocą polecenia STATUS, które wyświetla w oknie Shell wykaz wszystkich aktywnych procesów wraz z ich numerami.

Inne sygnały

AmigaOS przewiduje cztery sygnały, które możemy wykorzystać do komunikacji między procesami. Są oznaczone następującymi literami:

- C,
- D,
- E,
- F.

Każdy z nich ma oczywiście inne znaczenie, uzależnione od programu, który wykorzystuje sygnały. Wyjątkiem jest "C", który zawsze powoduje zatrzymanie.

Potoki

"Potoki" to rozwiązanie nieco podobne do urządzenia Ram Disk. Praca jest jednak w tym wypadku bardziej ukryta przez niepowołanym dostępem. Potoki funkcjonują bowiem jako urządzenie "PIPE:". Jest to kanał komunikacji między procesami. Zaletą potoku jest fakt, iż dane są przesyłane bezpośrednio z jednego procesu do drugiego. Gdybyśmy chcieli zrobić to samo wykorzystując plik tymczasowy w "RAM:", jeden

```
Workbench Screen
Workbench
Ram Disk
AmigaShell
4.Workbench3.0:> dir
Classes (dir)
Expansion (dir)
System (dir)
Rexxc (dir)
Utilities (dir)
L (dir)
S (dir)
T (dir)
C (dir)
Devs (dir)
Prefs (dir)
***Break
4.Workbench3.0:> status
Process 2: Loaded as command: C:\ConClip
Process 3: Loaded as command: Workbench
Process 4: Loaded as command: status
4.Workbench3.0:>
```



program musiałby czekać, aż drugi skończy swoje zadanie i zapisze cały plik, aby móc go odczytać. W przypadku urządzenia "PIPE:", odczyt jest możliwy także w trakcie zapisu danych. Dane możemy wysyłać bezpośrednio do urządzenia logicznego "PIPE:". Na przykład tak:

DIR SYS: >PIPE:

Możemy także z niego korzystać identycznie jak z dowolnego innego urządzenia:

DIR SYS: >PIPE:a

Prawdziwym zastosowaniem potoków jest przesyłanie bardzo dużych plików bezpośrednio z jednego programu do drugiego, bez konieczności oczekiwania na utworzenie pliku tymczasowego, jak musiałoby to mieć miejsce w "RAM:".

Komunikacja między zadaniami

Komunikacja między zadaniami może być realizowana w oparciu o potoki lub sygnały. Najczęściej jednak wykorzystuje się mechanizm portów. Port to jakby skrzynka kontaktowa między dwoma zadaniami, do której można wysłać dowolny komunikat. Praca większości rozwiązań w systemie Amigi jest oparta właśnie na portach. Użytkownik jednak ma ograniczoną możliwość ich tworzenia. Nie zrobimy tego za pomocą Shell. Musimy wykorzystać język ARexx, którego nauka wymaga poznania elementów programowania oraz kilku funkcji biblioteki Exec.

Komunikacja między programami

Dla użytkownika przeznaczony jest mechanizm komunikacji między aplikacjami. Opiera się on przede wszystkim o opisany niżej Clipboard oraz wykorzystanie języka skryptowego ARexx. Można wykorzystać także potoki, ale ich użycie jest trudniejsze i przeznaczone wyłącznie do przesyłania danych.

ARexx, jako język pozwalający na sterowanie aplikacjami, ma znacznie większe możliwości.

Clipboard

Clipboard (czyli Schowek) przypomina tablicę, do której możesz przypinać karteczki z informacjami. Każdy program AmigaOS ma możliwość kopiowania danych na Clipboard. Danymi mogą być fragmenty projektu, tekst itd. System sprawia, że dane wycięte z jednej aplikacji są dostępne w innej (patrz menu "Edycja" większości programów). Oczywiście, nie możesz wymagać niemożliwego. W opisany sposób nie wstawisz grafiki do zwykłego edytora tekstu, ale tekst do programu graficznego już tak. Za pomocą Clipboardu możesz między innymi:

- kopiować dane z Shell do edytora lub procesora tekstu i na odwrót (skorzystaj z kombinacji AMIGA + C, aby skopiować dane oraz AMIGA + V, aby je wstawić),

UWAGA: w oknie Shell możemy zaznaczać tekst tak, jak w edytorze, choć nie widać tego na pierwszy rzut oka.

- skopiować brush (wycinek) z programu graficznego do innego programu operującego na danych graficznych, na przykład systemowy IconEdit,

- przechowywać tymczasowe kopie dowolnych danych, bez względu na format.

Clipboard pozwala na integrację wielu aplikacji tak, jakby tworzyły one jedność. Do kopiowania, wycinania oraz wstawiania danych służą te same kombinacje klawiszy oraz te same opcje menu.

ARexx

ARexx jest uniwersalnym językiem programowania. Wyróżnia się maksymalną prostotą, dlatego każdy może nauczyć się przynajmniej pod-

staw tego języka. Najprostsze programy nie są bardziej skomplikowane od skryptów AmigaDOS.

ARexx pozwala na sterowanie dowolną aplikacją. Oczywiście musi ona pozwalać na wykorzystanie tego mechanizmu, czyli być wyposażona w tak zwany "port ARexxa". Wszystkie większe programy posiadają taką możliwość. Dzięki językowi ARexx wiele czasochłonnych i po prostu nudnych czynności komputer wykona sam. Użytkownik musi tylko uruchomić odpowiedni program (skrypt). Bardzo często nie trzeba nawet poznawać podstaw ARexxa, bo do wielu programów jest dostarczony zestaw skryptów automatyzujących wiele zadań, ponadto mogą one posiadać graficzne menu ułatwiające obsługę.

ARexx pozwala na integrację programów. Polega to na tym, że komunikując się poprzez skrypty, Clipboard oraz potoki, wiele poszczególnych funkcji programów może zostać wykorzystanych do wykonania skomplikowanego, wspólnego zadania tak, jakby stanowiły jedną dużą aplikację. Typowym przykładem jest adresowanie kopert przy użyciu procesora tekstu sprzężonego z dowolną ilością baz danych. Bazy sukcesywnie przesyłają adresy, procesor tekstu je drukuje sygnalizując, kiedy potrzebne mu będą kolejne dane. Użytkownik w tym czasie udaje się na zasłużony odpoczynek i wraca "na gotowe". ARexx pozwala również na tworzenie samodzielnych programów. Z jego poziomu dostępne są wszystkie ważne funkcje systemu operacyjnego. Wyjątkiem jest tworzenie graficznego interfejsu użytkownika, choć istnieją programy umożliwiające i ten proces. Wiele z nich znajdziemy na słynnym Aminecie. Oczywiście do takiego wykorzystania systemu potrzebna jest dobra znajomość mechanizmów drzemających w AmigaOS. Tym bardziej polecam studiowanie ogromnych możliwości tej platformy.

Adam Zalepa

re river's edge



Niezależne piśmiennictwo:

- recenzje płyt



- relacje z koncertów

- artykuły o Commodore 64

- strefa retro i hobby



- wywiady

- promocja artystów



- relacje z wydarzeń

- patronat medialny

Zajrzyj na stronę już teraz!

www.riversedge.pl

Kontakt z autorem:

pawel.riversedge@interia.pl

Lumberjack Reloaded



Wciel się w wesołego drwala zwanego Lumberjackiem, aby ściąć wielkie drzewo jednocześnie unikając spadających gałęzi.

Lumberjack traci zapał do pracy w miarę upływu czasu.

Kompletuj bonusy, aby grać dłużej i zdobywać coraz lepsze wyniki!

Wersja CD działa na Amidze CDTV i CD32!

Pozwala także zapisywać i odczytywać listy najlepszych wyników z pamięci NVRAM konsoli CD32.

Wymagania sprzętowe: 512 KB RAM, A500, 1200, CDTV, CD32