

C&A fan

AMIGA

- Historia Amigi cz.7
- A570... i na tym koniec?
- Sprawy techniczne i MegaRamHD
- Amiga Guide - szkółka
- LHA cz.2
- Asystent Pro 2.0
- Grimm
- Polskie Pismo Amigowe

Commodore 64

- Mimic Systems' Spartan
- Dual Drive Burst
- Wywiad: Romuald Drahokaupil
- Krótka historia kartridży Black Box
- Jak podłączyć kartridż do C-64 i jak przełączać pamięć?
- Przerabianie kartridży
- Rozszerzenie +60K
- Instalujemy +60K
- +32KB RAM Expansion dla stacji dysków 1541 II
- Programować w assemblerze każdy może
- Wektory na komodory cz.2
- Współczesne środowisko programowania na Commodore 64



W NUMERZE



maj 2010
numer 6 (1/2010)

WSTĘP

- 2 Od redakcji
- 3 News

C64

- Mimic Systems' Spartan 44
- Dual Drive Burst 47
- Wywiad: Romuald Drahokaupil 48
- Krótką historią kartridży Black Box 55
- Jak podłączyć kartridż do C-64 i jak przełączać pamięć? 57
- Przerabianie kartridży 59
- Rozszerzenie +60K 61
- Instalujemy +60K 62
- +32KB RAM Expansion dla stacji dysków 1541 II 64
- Programować w assemblerze każdy może - obroty 2D oraz rysowanie pikseli 65
- Wektory na komodory cz.2 69
- Współczesne środowisko programowania na Commodore 64 82

AMIGA

- 6 Historia Amigi cz.7
- 10 A570... i na tym koniec?
- 11 Sprawy techniczne i MegaRamHD
- 12 Amiga Guide szkółka
- 14 LHA cz.2
- 16 Asystent Pro 2.0
- 18 Grimm
- 20 Polskie Pismo Amigowe

GRY

- 22 FPS'y na Amigę - cz. 2
- 38 Pograjmy jak za dawnych lat... 1984
- 41 Booga-Boo
- 42 Tales of the Arabian Nights

C128

- 46 +60K dla C128

INNE

- C&A - widziane i zasłyszane 43
- Handlarze giełdowi 85
- Wywiad: Stein Eikesdal 87
- Hyde Park 94
- SILESIA AMIGA CLASSIC PARTY VOL. 5 95
- Listy 97
- Wasze opinie 98

Od Redakcji

Po ponad czteromiesięcznej przerwie mamy kolejne wydanie! Każdy nowy numer magazynu przynosi i zaskakuje nas czymś nowym - i tak też jest z **C&A Fan nr 6**. Tym razem jest więcej artykułów dla fanów komputera Amiga, co powinno ich ucieszyć. Oczywiście fani Commodore też znajdą coś dla siebie - zapraszam chociażby do przeczytania historii znanego kartridża **Black Box** oraz wywiadu z jego twórcą.

A co wydarzyło się w redakcji od czasu naszego ostatniego spotkania? Do jej składu po dłuższej przerwie powrócił Milek, który napisał do tego wydania dwa ciekawe artykuły. Od piątego numeru pisma, głównym jego składaczem jest Black Light, który stara się jak może, aby poświęcać na to każdą wolną chwilę. Korektą, oprócz artiego, zajmuje się również Atreus.

Strona magazynu znacznie się rozbudowała - do Waszej dyspozycji jest lepsze forum, a od czasu do czasu na stronie pojawiają się ciekawe artykuły, które z różnych przyczyn nie zostały opublikowane w magazynie.

Aby śledzić losy pisma **C&A Games**, również warto od czasu do czasu zaglądać na www.ca-fan.pl.

Na koniec tradycyjnie ponawiamy nasz apel. Nadal szukamy ludzi, którzy chcieliby z nami współpracować i pisać artykuły na różne tematy związane z komputerami firmy Commodore.

Wspomniane wyżej artykuły to nie jedyne, z jakimi warto się bliżej zapoznać. Amigowcy zapewne zechcą przeczytać kolejną część historii Amigi - tym razem traktuje ona o grach. Posiadaczom A500 próbujemy udowodnić, że zakup A570 nie zamyka możliwości rozbudowy ich komputera. Don Rafito dzieli się swoimi przemyśleniami na temat formatu Amiga Guide, a Mr Mat powraca do kursu LHA. Ci, którzy nie potrafią utrzymać w ryzach swojego portfela, powinni bliżej przyjrzeć się programowi Asystent Pro 2.0.

Czytając magazyn przekonacie się, że Grimm to nie tylko nazwisko utalentowanego rodzeństwa baśniopisarzy, ale także całkiem niezły program graficzny. Dla Commodorowców także nie zabrakło ciekawych materiałów. Możecie poczytać o rozszerzaniu pamięci, o bardzo szybkim kopierze dysków (Dual Drive Burst Backup), podajemy też mnóstwo informacji o kartridżach.

Axel udowadnia, że programować w assemblerze każdy może, a Wegi po raz drugi zagania wektory na komodory. O grach piszemy nieco mniej, niż w piątym numerze, ale też nie będziecie zawiedzeni. Nasz redakcyjny spec od rozrywki - p.a. 0 wspomina rok 1984 (a jest co wspominać!). To oczywiście nie wszystko, co znajdziecie w **C&A Nr 6**.

Zapraszamy do lektury!

Redakcja

Commodore & Amiga Fan

Magazyn użytkowników komputerów firmy Commodore

Redaktor naczelny: Ramos

Autorzy artykułów: Artykuły pisali: Axel, arti, Don Rafito, Romuald Drahokaupil, Klax, Milek, MrMat, p.a., Ramos, Skull, V-12, Wegi, YTM.
Założa FanCA: arti, Atreus, Bago Zonde, Black Light, Don Rafito, Indyjr, Milek, MrMat, p.a., Ramos, Skull, TOUDIIdel.

Korekta: arti, Atreus

Fotografia reportażowa: MrMat

Fotografia na okładce oraz w temacie numeru: Wikipedia, montaż Black Light
Desing, skład i łamanie: Black Light - procreators.pl

Newsy Newsy Newsy

Biblioteka Klasyki Komputerowej

Nie dawno powstała wersja beta strony nzwanej „Biblioteka Klasyki Komputerowej”, jest to kontynuacja wcześniejszych projektów m.in.: „CCOnline” czy „RetroReaders”. Na tej stronie znajdziecie wszystkie numery „Commodore & Amiga” jak i „C&A Fan”, a także masę innych czasopism, książek związanych z tematyką starych komputerów. Projekt ma być rozbudowany i można spodziewać się zmianę nazwy. Ma być powiązany ze stroną magazynu C&A Fan, który swoje archiwum tam umieści. Jednak co będzie, to czas pokaże. Obecnie trwają prace nad nowym wyglądem strony oraz nad jej udoskonaleniem.



Pograj na komciu w sieci

W styczniu pojawił się JSC64. To emulator Commodore 64 napisany w javascript przez Tim de Koning-a, czyli gotowy do umieszczania na własnych stronach WWW. Jest to tzw. port z produkcji Darron Schall i Claus Wahler napisanego w Actionscript-cie o nazwie FC64. Obecny emulator jest potwierdzeniem koncepcji możliwości wykonania tego typu programu w JavaScript-cie z wykorzystaniem biblioteki jquery oraz skorzystania z metod „canvas tag”, które są dostępne w najnowszych przeglądarkach t.j. Firefox, Chrome and Safari. Więcej informacji na stronie

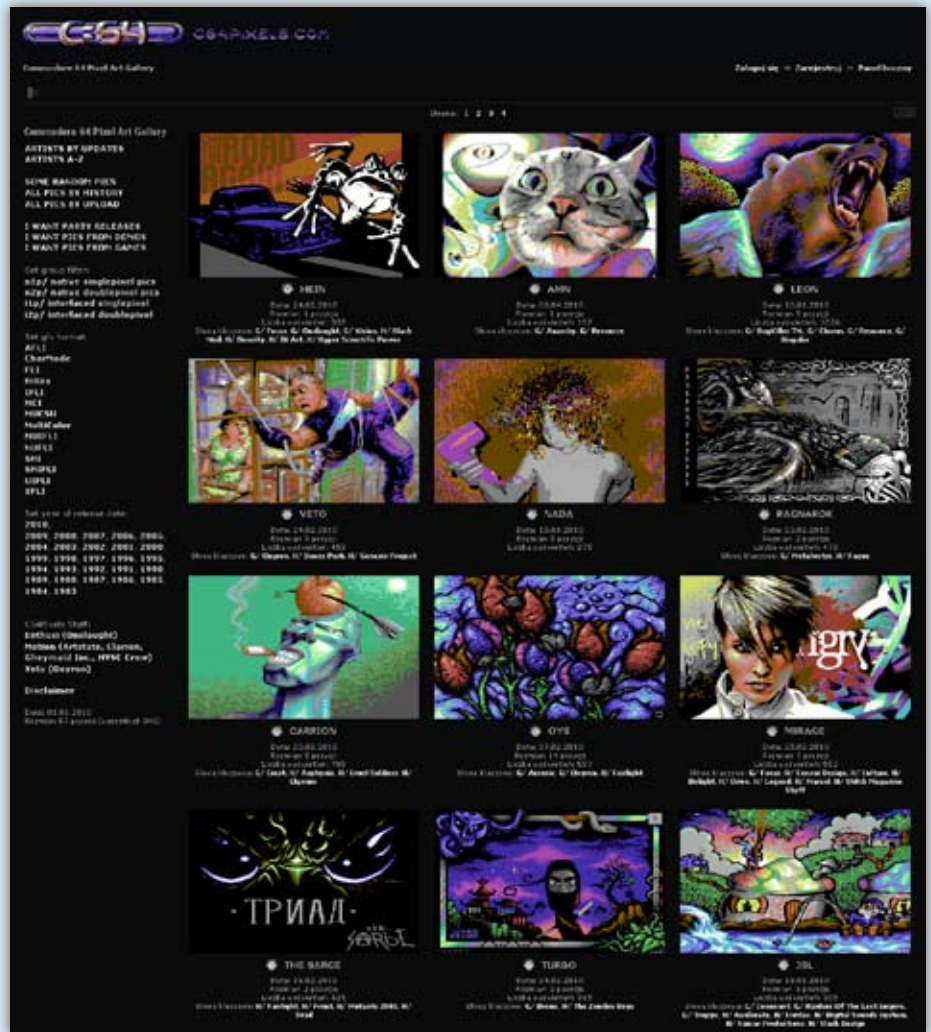
www.kingsquare.nl/jsc64.

C64Pixels.com

Od pierwszego stycznia powstała strona poświęcona grafice z komputera Commodore 64. Znajdziecie tam **mnóstwo obrazków w różnych trybach graficznych C64**, zapisanych jako pliki *.jpg, które są dokładnie podpisane i każdy kto szuka ulubionego trybu graficznego, grafika czy nawet grupy, znajdzie tutaj. Jest to obecnie jedyne archiwum obrazków z C64. Pod danym obrazkiem jest kilka istotnych informacji takich jak autor, tryb graficzny, data stworzenia

obrazka czy nawet skąd można ściągnąć oryginalny obrazek. Strona jest powiązana z archiwum CSDb, z której ściągamy większość oryginalnych obrazków.

<http://c64pixels.com/main.php>



HardSID Uno oraz HardSID UPlay

W lutym na stronie www.hardsid.com ujrzały światło dzienne dwa nowe „niskobudżetowe” produkty **HardSida**. **HardSID Uno** oraz **HardSID UPlay** - bo o nich właśnie mowa - to przenośne odtwarzacze utworów napisanych dla Commodore 64 pod jego sławny układ dźwiękowy SID. **HardSID UPlay** to wersja rozbudowana UNO, umożliwiająca odtwarzanie muzyki na starej lub nowej wersji chipa

Teraz ulubioną muzykę z dema lub gry, można zabrać ze sobą w każde miejsce - więcej info na:

www.hardsid.com/products.php

DATASTORM 2010

Między 5 a 7 lutym w szwedzkim Gothenburg-u odbyło się pierwsze większe party sceny komodorowskiej w roku 2010 o nazwie **Da-**

astorm. Party skupiło głównie posiadaczy c64 i Amig, dla których zresztą odbyły się pierwsze konkursy scenowe na najlepsze demo, muzykę oraz grafikę. Warto zajrzeć na stronę party www.datastorm.se i sprawdzić produkcje, bo na pewno każdy znajdzie dla siebie coś interesującego.

FOREVER 2010

W marcu już jedenasty w historii w słowackim Ternečinie odbyło się party **Forever**. Produkcje z tej imprezy można obejrzeć na stronie party lub w ogólnosiwiatowym serwisie demosceny c64 na CSDb. Relację z imprezy skrupulatnie opisał Sebaloz/Lepsi a można ją znaleźć w portalu poświęconemu polskiej scenie na c64 www.c64scene.pl.

Newsy Newsy Newsy

Wortal o C128

Posiadaczy C128 ucieszy nowina o nowo powstałej stronie dotyczącej właśnie tego modelu firmy Commodore. Temat C128 jest nadal w znacznym stopniu niewyczerpany, dlatego każda taka strona to cenne źródło wiedzy.



www.commodore128.host.sk

CCS64 v. 3.8

22 marca wyszła nowa wersja popularnego emulatora dla c64 CCS64 w wersji 3.8. Z nowości warto zwrócić uwagę na dwa ciekawe tryby wyświetlania emulacji - CRTtv oraz CRTmonitor. Są próbą odzwiedniczenia jakości wyświetlanego obrazu na starych odbornikach tv oraz monitorach lampowych. **Efekt jest dość ciekawy bo dodano specjalnie zakłócenia w obrazie!** Ma to symulować widok jak w latach osiemdziesiątych, ery „przedplazmolcd”-owej.

CCS64 v3.8 Changelog:

- FIX: Fixed a bug in the sprite-to-sprite collision-detection register.
- FIX: Fixed a bug with the 'Windows' menu Options -> Video behaviour.
- FIX: Made the graphics-drawing routines multi-threaded, for better performance.
- FIX: Fixed a bug which caused the QuickLoad/QuickSave features to hang.
- ADD: Two new Filter options, called CRT Television and CRT Monitor, which work only with the Window 2x screen mode and not with any other non-default video settings. These new Filter options include the after-lightening effect.
- INF: CCS64 V3.7 was compiled with Visual Studio 2008, which led to an in-compatibility with Windows 98/ME, but greater compatibility with Windows 7; when previous versions were compiled with Visual Studio 2005, this had the opposite effect.

www.ccs64.com

SEUCK Competition 2010

W marcu również poznaliśmy zwycięzców konkursu **Sideways SEUCK Competition 2010** organizowanego przez Richard Bayliss-a. Jest to konkurs organizowany od paru lat na najlepszą grę tzw. SEUCK-ową - napisaną pod ogólnodostępnym edytorem do gier. Sam organizator ma nie mały wkład w tego typu produkcje, a w tym roku do konkursu stanęło 8 prac, a każda z nich warta jest obejrzenia. Pierwsze miejsce zajęła gra „**Pour Le Merite**” której autorem jest Bamse.



BREAKPOINT 2010

Kwiecień również obfitował w imprezy scenowe z których największą był **Breakpoint 2010** skupiającą ludzi ze sceny praktycznie ze wszystkich platform. Koniecznie trzeba obejrzeć produkcje wystawione w „kompotach” gdyż niektóre były wręcz rewolucyjne.

Dokładny opis imprezy wykonał niezmordowany w tej materii Sebaloz/Lepsi i również można go przeczytać na www.c64scene.pl. Raport z tej imprezy znajdziemy również na riversedge.pl.

STARY PIERNIK 5

Warto odnotować, że w marcu odbyło się pod Toruniem party polskiej części sceny c64 - mowa tu oczywiście o **Starym Pierniku 5**. Relację na www.c64scene.pl tym ra-

zem opisał Bimber/Arise, a impreza odbyła się szerokim echem po świecie z powodu dość kontrowersyjnego dzieła ukazanego na imprezie. „**Pro Memoria 3 (96%)**” bo taką nazwę miała ta już słynna produkcja, to demo wyświetlone na party, a potem.... spalone.

Oprócz uczestników imprezy nikt nie widział go później, dlatego też po wstąpieniu produkcji do bazy demosceny (CSDB), rozpoczęły się protesty. Na szczęście inne produkty nie zostały spalone i można je oglądać do woli.

HACKERS TOP 2010

Pod koniec kwietnia miała miejsce jeszcze jedna impreza, która odbyła się u naszego wschodniego sąsiada na Białorusi. **Hackers Top 2010** party, był imprezą multiplatformową, a do nowości można zaliczyć sposób głosowania w poszczególnych konkurencjach - poprzez sieć, a więc na ogólny wynik konkursów miały osoby nie tylko uczestniczące bezpośrednio w imprezie.

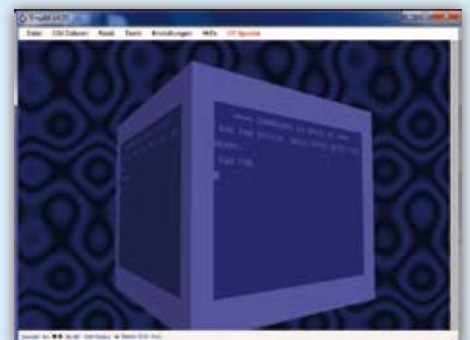
<http://party.c64.su/ht2010/>

Nowy EMU64

Na początku maja ukazała się nowa wersja, dość świeżego koncepcyjnie emulatora c64 o nazwie **EMU64**. Trzeba przyznać, że emulator jeszcze nie jest szeroko stosowany, ale szybkość jego rozwoju (**obecna wersja to 4.3 !**) może niejednemu zaimponować, oraz dobrze rokuje na przyszłość.

Emulator jest mocno robudowywany na wszelkiego rodzaju rozszerzenia dostępne na c64 - na tym polu konkuruje z sukcesem z innymi obecnymi. Niemieccy autorzy pokusili się nawet o tryb „fun” w którym to emulacja wyświetlana jest na obracającym się szkieletku. Do ściągnięcia ze strony

www.emu64.de



Newsy Newsy Newsy

PPA Nr 2

W bieżącym numerze C&A Fan możecie przeczytać recenzję pierwszego numeru „Polskiego Pisma Amigowego”. Tymczasem na stronie ppa.pl można już zamówić kolejne wydanie. Czy warto? Naszym zdaniem TAK. Znajdziecie tam kontynuacje rozpoczętych artykułów, rozpoczęcie nowych, publicystykę i wiele innych rzeczy. Cena: 20 zł.



www.ppa.pl/magazyn/

Powrót Commodore?

Oczywiście nie chodzi o nową wersję kultowego C64. Oto **Commodore Phoenix**. Co widzicie na zdjęciu wprowadzie mocno przypomina naszego ulubionego 8-bitowca, ale w swoim wnętrzu kryje płytę główną z chipsetem

Intel G31 Express ze zintegrowaną kartą graficzną Intel Graphics Media Accelerator 3100, kartę dźwiękową Realtek AL888 HD oraz siecią Gigabit Ethernet. Obudowa wykonana z anodowanego aluminium została opatrzona logiem C64. Szczegółowa konfiguracja zależy od użytkownika. Może zawierać procesor Intel Core 2 Duo lub Core Quad, do 4 GB pamięci DDR2, dysk twardy o pojemności nawet 2 TB oraz napęd optyczny. Cena Commodore Basic „Bare Bones” (zapewne w wersji podstawowej) to 395 dolarów. Ma być dostępny od czerwca 2010 r.



http://www.commodoreusa.net/Commodore_Phoenix_computer.html

Nowe sterowniki dla kart NVIDIA

Pojawił się sterownik 2D do kart opartych na **NVidii dla AROS'a**. Sterownik ten został napisany w celu sprawdzenia, czy posiadana karta odpala się w systemie. Obsługuje karty z serii GF1 do GF7. Pozwala na wybranie jednej z trzech rozdzielczości (640, 800, 1024).

Sterownik nie posiada akceleracji, jest bardzo wolny i prawdopodobnie mało stabilny. Do pobrania w poniższym linku

<http://www.amiga-news.de/en/news/AN-2010-05-00029-EN.html>

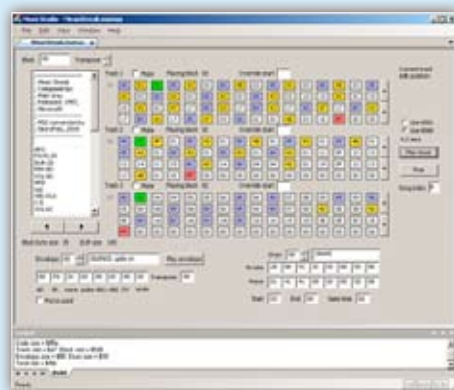


Aktualizacja Amiga Forever 2009

Pojawiła się darmowa łątka do pakietu „**Amiga Forever 2009**”. Rozwiązuje ona problemy takie jak: automatyczne przywracanie skasowanych katalogów, kopiowanie plików *.PDF z RP9, pokazywanie Update'ów, itp.

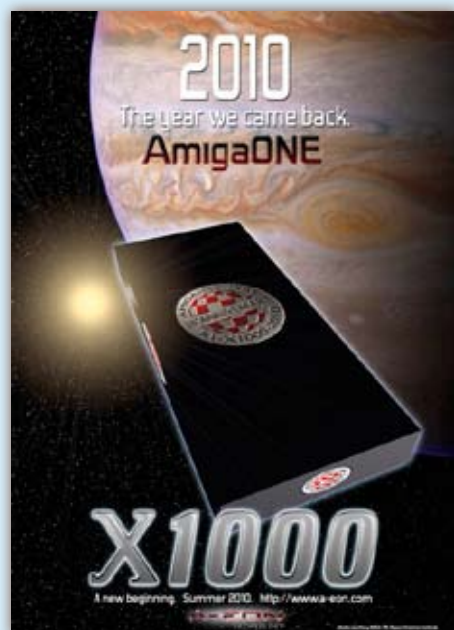
www.amigafuture.de/viewtopic.php?t=26423

Music Studio 2.0.0.16 - edytor SID na PC



<http://noname.c64.org/csdb/release/?id=91544>

AmigaOne - nadchodzi nowa Amiga



<http://www.a-eon.com/>

HISTORIA AMIGI cz. 7:

Gra toczy się dalej!

Najpotężniejsza platforma do gier

Amiga rozpoczęła swoje życie jako wyspecjalizowana maszyna do gier i mimo że bardzo szybko wyrosła na pełny komputer, nigdy nie straciła rozrywkowej strony. Paleta 4096 kolorów, sAMPLEowany dźwięk stereo i chipy do akceleracji grafiki uczyniły z tej maszyny idealną platformę do gier i firmom związanym z tą branżą nie zabrakło czasu rozpoczęcie korzystania z tej mocy.

Niskie wyniki sprzedaży Amigi 1000 ograniczały liczbę gier, które deweloperzy chcieli napisać dla tej platformy, ale kiedy Commodore wypuścił podstawowy model, Amigę 500 w 1987, wszystko się zmieniło. Teraz najpotężniejszy komputer do grania był także jednym z najtańszych, a firmy tworzące gry rzuciły się na okazję zaprezentowania swoich talentów na Amidze.

Mind Walker (1986)

Jedną z pierwszych gier wydanych na Amigę była raczej dziwaczna perełka, zwana *Mind Walker*, napisana przez Billa Williamsa i opublikowana przez samo Commodore. Williams rozpoczął swoją karierę w projektowaniu gier na Atari 800, pisząc takie klasyki, jak *Necromancer* czy *Alley Cat*. Jego gry były zawsze wyjątkowe, łączyły interesujące sytuacje z innowacyjnym sposobem grania.

Mind Walker umieszczał gracza w roli fizyka, który postradał zmysły. Zamiast uciekać się do leków lub terapii, bohater gry zdecydował się wysłać swoje podzielone ego do głębin swego własnego mózgu. Zadaniem gracza było przejście tego surrealistycznego krajobrazu i odkrycie ścieżek prowadzących w coraz to głębsze poziomy, a ostateczny cel to odnalezienie ukrytego klucza do uratowania własnego zdrowia psychicznego.

Twoje alter ego skacze po jaskrawych, kolorowych kwadratowych platformach na różnych wysokościach – na szczęście nie można spaść. Jeżeli dojdiesz do końca ekranu, tam natychmiast ładuje się przyległy obszar. Unoszące się złote kule próbują zestrzelić cię

za pomocą śmiertelnych reflektorów, jednak możesz odstrzelić większość z nich, używając ładunków energii elektrycznej, którą kierujesz ze zręcznością Lorda Sith. Nad niektórymi kwadratami unoszą się dziwne piramidy, które zamieniają twoją postać z człowieka w czerwonego czarodzieja, w latającego, podobnego do robaka, obcego lub w seksowną uwodzicielkę. Te różne formy muszą odnaleźć różne części ścieżki, a śledzenie całej operacji wymaga dokładnej konsultacji z umieszczoną na górze mapą. Kiedy bohater jest schowany za zbyt wysoką platformą, gracz może przełączać się pomiędzy jednym z czterech różnych widoków za pomocą liter N, S, E lub W.



Mindwalker firmy Commodore

Kiedy dana ścieżka jest skończona, gra przełącza się na trójwymiarowy widok psychodelicznego tunelu. Gracz, za pomocą swoich bezcielesnych rąk, musi chwycić przezroczyste zielone drzwi, które prowadzą do ostatecznego poziomu, w którym to bohater broni się przed niewyraźnie wyglądającymi „złymi myślami”, by odnaleźć kolejny kawałek swojej poczytalności.

Gra posiada prostą, lecz sugestywną grafikę, robiącą dobry użytek z wbudowanego w Amigę sprzętu do rysowania wielokątów i wypełniania obszarów. Bill Williams, zanim zajął się projektowaniem gier, był kompozytorem, a muzyka, którą stworzył dla *Mind Walkera* miała niesamowitą, liryczną barwę, co doskonale pasowało do tematu gry. Wykorzystany został efekt stereo, dzięki czemu wydaje się, że pioruny przepalają się przez cały pokój. Tak jak sama Amiga, *Mind Walker* był grą niezwykłą i skłaniającą do refleksji.

Kolejną niezwykłą cechą tej gry było to, że nie dość, iż mieściła się ona na jednej dyskietce, to jeszcze nie miała zabezpieczenia przed kopiowaniem i można ją uruchomić bezpośrednio z GUI Workbench Amigi. Co więcej, gra może korzystać z wielozadaniowości, więc można bez problemu uruchamiać

inne aplikacje w tle. Niewiele amigowych gier w przyszłości miało zachować te cechy. Twórcy gier, pragnąc wycisnąć ostatnią kroplę mocy z komputera, woleli omijać system operacyjny i mieć bezpośredni dostęp do sprzętu. Dzięki temu późniejsze tytuły były bardziej imponujące pod względem graficznym, ale za cenę rezygnacji z wielozadaniowości.

Bill Williams będzie kontynuował pisanie gier aż do 1992 roku, kiedy ingerencje firmy w tytuł *Bart's Nightmare* – „Koszmar Barta” (sam twórca nazywał go „Bill's Nightmare” – „Koszmar Billa”) dla Super Nintendo stały się powodem, dla którego opuścił przemysł i oddał się drugiej karierze – luterańskiego pastora, po drodze zdobywając tytuł magistra teologii.

Defender of the Crown (1986)

Firma Cinemaware została założona przez Roberta i Phyllis Jacobów w 1985 roku. Ich celem było tworzenie gier, których styl i prezentacja przywodziłyby na myśl filmy. Był to cel ambitny w czasach, kiedy większość gier wideo to proste strzelanki albo labiryntówki, lecz nadejście Amigi dało niewielkiej firmie szansę na realizację tych marzeń.

Gra *Defender of the Crown* była ich pierwszym tytułem i pokazywała moc nowej platformy. Scena: jesteś saksońskim baronem na angielskim lennie w średniowieczu, a król dopiero co został zabity bez wyraźnego następcy. Musisz walczyć z innymi Sasami i z normanńskimi najeźdźcami, podbić Anglię i stać się nowym królem. Ta gra była jedną z pierwszych, które posiadały przepiękne, ręcznie malowane ekrany, wyświetlane podczas wczytywania i wprowadzające w akcję, zaś sama gra była równie piękna. Każda runda rozpoczynała się od stylizowanego widoku Brytanii z lotu ptaka. Z tego menu gracz mógł wybrać atak na sąsiednie księstwo, rozpocząć atak na zamek wroga, urządzić turniej rycerski albo też sporadycznie odegrać śmiałą akcję ratowania pięknej dziewczyny. Od czasu do czasu pojawia się Robin Hood (NPC*), który może być zarówno wrogiem, jak i sprzymierzeńcem.

Tak jak w wielu grach tamtej epoki, osiągnięcie zwycięstwa mogło być frustrującym

trudne. Przykładowo na ekranie z najazdem kontrolujesz pojedynczego wojownika, który musi wycinać wroga po wrogu, podczas gdy jego krajanie jedynie trzymają resztę wrogów z daleka. Turniejowe potyczki są tylko nieco mniej skomplikowane niż w prawdziwym życiu, wymagają trzymania na myślenie pewnej ręki, która umieści rycerską kopię we właściwej pozycji we właściwym momencie. Zwycięstwo w turnieju może przynieść ci punkty honoru albo nawet terytorium, w zależności od początkowej stawki.



Defender of the Crown firmy Cinemaware

Defender of the Crown na początku była grą wyłącznie dla Amigi i często dealerzy używali jej do prezentowania platformy entuzjastycznym młodym graczom. Wiele uznania za sukces gry należy się jej artyście, Jimowi Sachsowi. RJ Mical, który pracował przy tej produkcji jako konsultant, wspominał jego talent.

„Jim Sachs, on jest bogiem” – zdumiewał się Mical. – „Jim Sachs jest niesamowity. Dziś wszędzie jest taka grafika, ponieważ teraz jest mnóstwo naprawdę dobrych grafików-artystów komputerowych, ale wtedy, 20 lat temu, to było niesamowite mieć kogoś tak dobrego.”

Ponieważ Cinemaware była początkującą firmą, której brakowało gotówki, a Defender był ich pierwszym produktem, zmuszeni byli wydać grę zanim ją całkowicie ukończono. Później porty gry dla Nintendo Entertainment System, Commodore 64, IBM PC i Atari ST wypełniły te brakujące części, wliczając w to bardziej treściwe ekrany z atakującą armią. Jednakże konwersje nie dostarczały takiej samej jakości dźwięku i grafiki, jak w wersji dla Amigi.

Cinemaware kontynuowała wydawanie innowacyjnych gier do roku 1991, kiedy to przeładowany i mocno opóźniający się tytuł o tematyce zimnej wojny – S.D.I. – zmusił firmę do ogłoszenia bankructwa. Dwóch wczesnych pracowników firmy, Lars Fuhrken-Batista i Sean Vesce, zeszło się razem, aby stworzyć udoskonaloną wersję gry, zwaną Robin Hood: Defender of the Crown na PlayStation 2, Xbox i Windows w 2003 roku.

Gry RPG przychodzą do Amigi

Faery Tale (1986)

Faery Tale to jedna z tych gier, które każdy, kto w nie grał, zapamiętuje. Osadzona w świecie fantasy gra role-playing stworzona przez MicroIllusions, która cechowała się widokiem z góry, Faery Tale przypomina klasyki, takie jak oryginalna Legend of Zelda albo seria Ultima i zawiera przy tym zaskakującą głębię.

Grę rozpoczyna przedstawienie głównych bohaterów poprzez wirtualną książkę, w której wolno obracają się strony. Trzech braci, Julian, Philip i Kevin, dorastało w małej wiosce Tambry w kraju Holm i teraz pragną odkryć szerszy świat. Julian, najdzielniejszy z trójki, wyrusza pierwszy. Świat, tak jak w wielu grach RPG, jest w niewytłumaczalny sposób pełen bandytów, potworów i nieumarłych stworzeń, takich jak szkielety. Często atakują oni w grupach, które mogą łatwo przytłoczyć postać gracza, zwłaszcza przy jego początkowym uzbrojeniu w postaci niewielkiego sztyletu. Akcja rozgrywa się w czasie rzeczywistym, bez tur albo pauz, a przetrwanie wczesnych etapów gry może być trudne.



Faery Tale Adventure firmy MicroIllusions

Kiedy Julian umiera, mała wróżka go wskrzesza, ale po kilku zgonach staje się on duchem. Gracz przeobraża się wówczas w Philipa, który może rozmawiać z duchem Juliana i odzyskać rzeczy z jego ciała. Jeżeli Philip zawiedzie, wyprawa podejmowana jest przez Kevina, będącego ostatnią nadzieją gracza. Na szczęście grę można zapisać w dowolnym momencie.

Postacie mają różne statystyki, które można poprawiać w miarę upływu czasu i treningów, tak jak w wielu grach RPG. Dzielność (Bravery) odzwierciedla siłę gracza, Żywotność (Vitality) – jego punkty życia (które zwiększają się w wolniejszym tempie, w miarę jak wzrasta Dzielność). Są również punkty Dobroci (Kindness), wymagane przy rozmowach z różnymi postaciami NPC. Inaczej niż w wielu grach role-playing, Faery Tale pozwala graczowi atakować niewinne postacie NPC, ale ponieważ pozostają one wówczas martwe, rzadko jest to dobrym pomysłem. Gracz musi również upewnić się, że zapakował wystarczającą ilość jedzenia na swoją długą podróż, ponieważ głód będzie stopniowo drenował jego Żywotność.

Obiekty na ziemi mogą zostać podniesione, a uzyskane skarby można wymienić na lepszą broń w lokalnym sklepie. Kiedy gracz zostaje wyposażony w broń, jest to natychmiast widoczne na jego postaci. Niektóre rzeczy mogą mieć charakter magiczny, jak Ptasi Totem (Bird Totem), który umożliwia graczowi widok z lotu ptaka na mapę. Są także kolorowe klucze, otwierające różne zamknięte drzwi, mikstury przywracające zdrowie a nawet cacka pozwalające na chwilę zatrzymać czas w ferworze walki.

Świat gry jest zdumiewająco duży i zawiera wiele niespodzianek i zwrotów akcji, jak choćby wielki żółw, którego gracz może użyć do transportu własnej osoby przez wodę. Później, aby uratować królewską córkę przed straszliwym przeznaczeniem, gracz musi ośwoić złotego łabędzia i przelecieć na nim nad niedostępny inaczej pasmem górskim.

Pomimo dość przyziemnej fabuły fantasy (gracz musi zgromadzić pięć złotych posągów, aby otworzyć portal do Świata Astralnego i pokonać złego Nekromantę), gra jest wciąż pamiętana po ponad dwudziestu latach. Rozmawiałem ze swoim przyjacielem, Domenico DiTomaso, który pamiętał o spędzeniu dwóch szczęśliwych miesięcy na graniu wspólnie z kolegą, aby ukończyć grę. „Mogłeś pójść w dowolne miejsce” – mówił, zdumiewając się otwartością świata gry, która pozwalała graczowi na wędrowanie przez cały świat bez nagle pojawiającego się ekranu ładowania. „Tylko pamiętaj” – powiedział – „upewnij się, że zrobiłeś mapę, kiedy wejdiesz do Jaskini Smoka!”

Dungeon Master (1988)

Dungeon Master właściwie miał swój debiut na zniechęconej platformie Atari ST na rok przed tym, jak wydano go na Amigę. Z uwagi na ten fakt, jakość grafiki była nieco gorsza niż Amiga była zdolna wytworzyć, ale widok 3D z perspektywy pierwszej osoby sprawił, że gra ta wyróżniała się od konkurencji. Choć grafika w przeprojektowanej grze pozostała w znacznej części niezmieniona, wersja Amigowa zrobiła dobry użytek z dedykowanych chipów dźwiękowych. Dźwięk stereo sprawił, że krzyki potworów zdawały się „wyskakiwać” w trzech wymiarach, co było ważną wskazówką, jeśli weźmiesz pod uwagę fakt, że wrogowie mogą podkraść się do gracza z dowolnego kierunku. Ta przewaga rzeczywiście przyczyniła się do lepszej sprzedaży Amig 500 niż Atari ST.

Dungeon Master był inspirowany surową trójwymiarową grafiką, pojawiającą się w serii Ultima, gdy gracz wchodził do lochów. Umieszczając całą grę w takim otoczeniu, twórcy z FTL Games mogli skupić się na udoskonalaniu grafiki 3D i samej grywalności.

Gra rozpoczynała się przed wejściem do lochu, z jedynym możliwym kierunkiem ruchu: do środka! Na pierwszym poziomie bezcielesny gracz kręcił się po „Sali Mistrzów”, z wieloma różnymi portretami bohaterów powieszonymi na ścianach. Zbliżenie się do portretu powodowało, że jego bohater w magiczny sposób pojawiał się jako część twojej drużyny: mogłeś mieć w sumie cztery postacie. Inaczej niż w innych grach tego typu, nie było tu innego procesu kreowania postaci: brałeś predefiniowanych awanturników takimi, jakimi są. Kiedy byłeś gotowy, schodziłeś na dół...



Dungeon Master firmy FTL

Wszystkie działania mogły być kontrolowane za pomocą myszki, od obracania się i poruszania, po podnoszenie obiektów. Kliknięcie na danym obiekcie przesunęło go do pustej dłoni aktualnie wybranej postaci; trzeba było otworzyć ekran z posiadanymi przedmiotami by przesunąć obiekt do plecaka. Nadmiar inwentarza mógł być wyrzucony za pomocą prawego przycisku myszki i mógł polecieć do przodu przez loch. Było wiele zagadek, ukrytych dźwigni i tajnych drzwi do otwarcia. Wskazówki można było czasami znaleźć w notatkach, rozrzuconych na wyższych poziomach lochu.

Walki miały miejsce w czasie rzeczywistym, a gracz musiał ręcznie przełączać się pomiędzy postaciami, żeby wziąć zamach czy rzucić czar na potwora. System rzucania czarów był dość innowacyjny: na przykład aby rzucić ognistą kulę, gracz mieszał symbol ognia z symbolem skrzydła. Kiedy jedna z postaci umierała (zdarzało mi się to na początku, gdy wpadałem w zapadnię) można było podnieść jej kości i zanieść do komnaty odrodzenia.

W Dungeon Master było w sumie 14 poziomów, a zakończenie ostatniego z nich wymagało zabicia władcy demonów, zwanego Chaosem, który wyglądał jak skrzyżowanie Dartha Vadera z Amadeuszem. Chaosu nie dało się zabić za pomocą zwykłej broni i trzeba go było uwięzić w magicznej klatce, zanim mógł zostać dobity. Fabuła – napisana przez Nancy Holder, pisarkę, która od tamtej pory pisała książki z serii „Buffy - Postrach Wampirów”, „Sabrina - Nastoletnia Czarownica” czy „Tajemnice Smallville” – pojawiała się w sa-

mej grze i szczegółowo przedstawiona została w podręczniku do niej. Dungeon Master zainspirował rzesze naśladowców, jak choćby Eye of the Beholder czy Captive. Był również główną inspiracją dla przełomowego arcydzieła 3D, jakim była gra Ultima Underworld.

Pionowe strzelanki też

Xenon II (1989)

Xenon II był kontynuacją popularnej pionowej strzelaniny, napisanej przez spółkę fanów Amigi, zwaną The Bitmap Brothers. Sequele świetnie wychodziły Bitmap Brothers; ich druga wersja futurystycznej piłki ręcznej Speedball była ogromnym sukcesem komercyjnym i jest czule wspominana aż po dzień dzisiejszy.

Przed nadejściem strzelanek 3D jednym z bardziej popularnych typów gier były dwuwymiarowe skrolowane strzelaniny, których akcja zazwyczaj umieszczana była w kosmosie. Gracz sterował pojedynczym statkiem, przeciwstawiając się nieskończonej flocie wrogich jednostek, które nie mogły zbyt szybko strzelać. Pierwsze automaty do gier miały ograniczoną moc obliczeniową i zazwyczaj umieszczały gracza na prostym tle gwiazd. Późniejsze gry miały bardziej szczegółowe tło, które samo w sobie mogło stanowić przeszkodę.

Tła i wrogowie w Xenon II były w znacznej części inspirowane mega hitem z automatów – R-Type, który stawiał gracza przeciwko dziwnym i poniekąd obrzydliwym szykom kosmicznych robali i innych paskudnych stworzeń. Xenon II miał i kosmiczne robale, i gigantyczne kosmiczne trylobity, i wreszcie bardziej standardowo wyglądające statki kosmiczne wroga. W przeciwieństwie do R-Type, który był skrolowany od prawej do lewej, Xenon II utrzymał zgodne ze starą szkołą skrolowanie pionowe. Z jedną różnicą: gracz mógł „przewinąć się” do tyłu na krótkim odcinku.



Xenon II firmy The Bitmap Brothers

Na końcu każdego poziomu gracz miał możliwość odwiedzenia sklepu, prowadzone-

go przez nieznośnego starego kosmitę. Dawał on pewne wskazówki odnośnie każdego dostępnego do nabycia gadżetu, ale jeśli naprzykrzałeś mu się zbyt długo, odburkiwał: „No co, może jeszcze chcesz, żebym za ciebie zagrał?”.

Xenon II nie zawierał żadnych błyskotliwych innowacji, nie przeddefiniował gatunku dwuwymiarowych skrolowanych strzelanek, pokazał jednak, że Amiga była zdolna, by w domu dostarczać wrażeń porównywalnych z arcadowymi maszynami do gier.

Shadow of the Beast (1989)

Podczas gdy do tej pory większość gier dla Amigi była lepsza niż porty do innych komputerów, wciąż nie było gry, która w rozstrzygający sposób mogła zdmuchnąć konkurencję, nie pozostawiając cienia wątpliwości, która platforma do gier jest najlepsza.

To znaczy tak było, dopóki Psygnosis nie wydał Shadow of the Beast. Jako platformówka z widokiem z boku w stylu Super Mario Bros., Shadow of the Beast popchnęła chipset graficzny Amigi aż do granic jego możliwości.

Jeszcze przed technologią graficzną 3D, gry z widokiem z boku często używały techniki zwanej przewijaniem równoległym (ang. parallax scrolling), gdzie obrazy w tle przewijane są wolniej niż te z pierwszego planu, co dawało złudzenie poruszania się w większym świecie. Niewiele konsol w tamtych czasach miało moc wystarczającą, aby w ogóle skrolować tło (Super Mario miał tła statyczne), ale kilka gier z automatów miało dwa lub trzy poziomy przewijania równoległego. Shadow of the Beast miał ich do dwunastu.

Także wrogowie nie byli zrobieni niedbale. W przeciwieństwie do niewielkich sprite'ów w innych grach z widokiem z boku, potwory w Shadow of the Beast mogły wypełniać do połowy ekranu.

Shadow of the Beast miał także intrygującą historię w tle. Bohaterem gry był człowiek o imieniu Aabron, który jako dziecko został porwany przez złą bestię – lorda Maletotha i zamieniony przez złą magię w strasznego człowieka-bestię, by służyć swemu nowemu panu. Kiedy to stworzenie jest świadkiem egzekucji człowieka, przypomina sobie swego ludzkiego ojca i zalewają je wspomnienia z dzieciństwa. Uciekając od Maletotha jest zdecydowany, by poszukać swojej zemsty.

Ukończenie 12 poziomów gry było frustrująco trudnym zadaniem. Bestia, mimo że sama w sobie potężna, wydawała się być stale na krawędzi śmierci. Nie chodzi tylko o inne potwory, z którymi trzeba było sobie poradzić, ale Bestia stawiała wobec niekończącej się zapory śmiertelnych pułapek z kolców wyrastają-



Shadow of the Beast firmy Psygnosis

cych z ziemi, latających szwadronów nabitych kolcami kul, a nawet olbrzymich unoszących się oczu. Gracz zaczynał z 12 jednostkami zdrowia, a każde dotknięcie wroga zmniejszało tę rezerwę o jedną jednostkę. Kiedy zdrowie spadało do zera, gra była skończona.

Grafika nie była jedyną częścią gry, która ją wyróżniała. „To, co przede wszystkim pamiętam w Shadow of the Beast, to muzyka” – mówi właściciel Amigi, Narendar Ghangas. „Przez cały czas grze towarzyszyło poczucie, że zbliża się coś złego i ponure instrumenty naprawdę pasowały do mrocznej natury tej gry. Pamiętam, że byłem totalnie zniewolony syntezowaną muzyką – to było jak nawiedzenie.”

Podczas gdy niektórzy narzekali, że w Shadow of the Beast cukierkową grafikę postawiono ponad głębię zabawy, sama gra była sukcesem handlowym i przeportowano ją później na platformy takie, jak Sega Genesis (bez wielu kolorów i kilku poziomów przewijania równoległego). Gra doczekała się również dwóch sequeli, z których ostatni wydano wyłącznie na Amigę.

Lemmings (1991)

Gdyby istniała jedna gra, która mogłaby opisać doświadczenie z Amigą, to musiałyby być to Lemmings. Gra wypuszczona przez Psygnosis w 1991 roku była dziwaczna, zabawna i uzależniająca. Gracze sterowali dużą liczbą ubranych na kolorowo, zielonowłosych lemingów, które potrzebowały pomocy w przedostaniu się od początku do końca każdego poziomu.

Bez pomocy użytkownika biedne lemingi poszłyby na urwisko wprost ku swojej zagła-

dzie. Na szczęście gracz mógł, klikając myszką, zlecać pewnym lemingom ustalone „prace”. Jednym z ważniejszych był leming „zatrzymujący”, który stał z rozpostartymi rękami i potrząsał głową tam i z powrotem, sprawiając, że inne lemingi, kiedy na niego wpadały, szły w przeciwnym kierunku. Dzierżący parasolkę leming miękko opadał w dół na ziemię, zamiast spadać prosto ku swej śmierci. Była nawet opcja leminga samobójcy, który odliczał od pięciu do zera, pisał „Oh no!” i wtedy eksplodował. Czasami takie poświęcenie było niezbędne, czasami było tylko świetną zabawą.



Lemmings firmy Psygnosis

Innym lemingom można było zlecać kopanie albo budowanie ramp, by pomóc reszcie grupy osiągnąć niedostępne lokalizacje. Posiadanie tych wszystkich opcji uczyniłoby ukończenie dowolnego poziomu błahym ćwiczeniem, ale był też pewien haczyk: każdy poziom dawał graczowi ograniczoną liczbę prac do wykorzystania i nie wszystkie prace były dostępne na wszystkich poziomach. Kiedy użytkownik był naprawdę sfrustrowany, zawsze pozostawała opcja „nuklearna”: zlecenie wszystkim lemingom naraz odliczania od pięciu do zera. Otrzymany w rezultacie tego chór tych wszystkich „Oh no!” i późniejsze całkowite zniszczenie były dziwnie oczyszczające.

Gra Lemmings była niewiarygodnie popularna i stała się pewnego rodzaju symbolem amigowej społeczności. Gail Wellington, szef Commodore Advanced Technical Support (CATS), zaplanował raz, żeby cała grupa pracowników Commodore ubrała się jak lemingi na pokaz handlowy. O wszystko zadbał: o liłowe ubrania, zielone włosy, odpowiednie pozycje, parasolkę, a nawet o balony wypełnione konfetti do nieuniknionego finału „Oh no!”. Trochę się niepokoił o tę ostatnią część: co o czymś takim pomyślą ludzie, którzy muszą sprzątać po takiej imprezie? Okazało się, że ich obawy były nieuzasadnione: personel sprząający nie miał nic przeciwko uprzątnięciu bałaganu po tak doskonałej rozrywce.



Przebrani pracownicy Commodore. Zdjęcie dzięki uprzejmości stevex.

Podczas gdy Lemmingsi były portowane na inne platformy, w szczególności na IBM PC, wersja amigowa miała najlepszy dźwięk, a nawet kilka niedostępnych nigdzie indziej opcji zabawy: dwóch graczy mogło grać równocześnie używając myszki, a to dzięki unikalnej zdolności Amigi, do której można było podłączyć w tym samym czasie dwa gryzonie.

W roku 2007 firma Sony wydała nową wersję Lemmingsów na swoją konsolę PlayStation Portable (PSP). Mając do wyboru wiele portów, jako podstawę dla gry zdecydowali się użyć kodu Amigi. Teraz całe nowe pokolenie może doświadczać radości pomiatania małymi zielonowłosymi stworzeniami.

Nie przegapcie następnej części cyklu, w której przyjrzymy się życiu kilku najpopularniejszych twórców gier, takich jak Team17, Psygnosis czy Bitmap Brothers.

Jeremy Reimer
Ars Technica
tłumaczenie: arti
korekta: KT (thx, Bro!)

*NPC (non-player character) – postać, w którą nie wciela się żaden z graczy, bohater niezależny (przyp. tłum.)

A570...

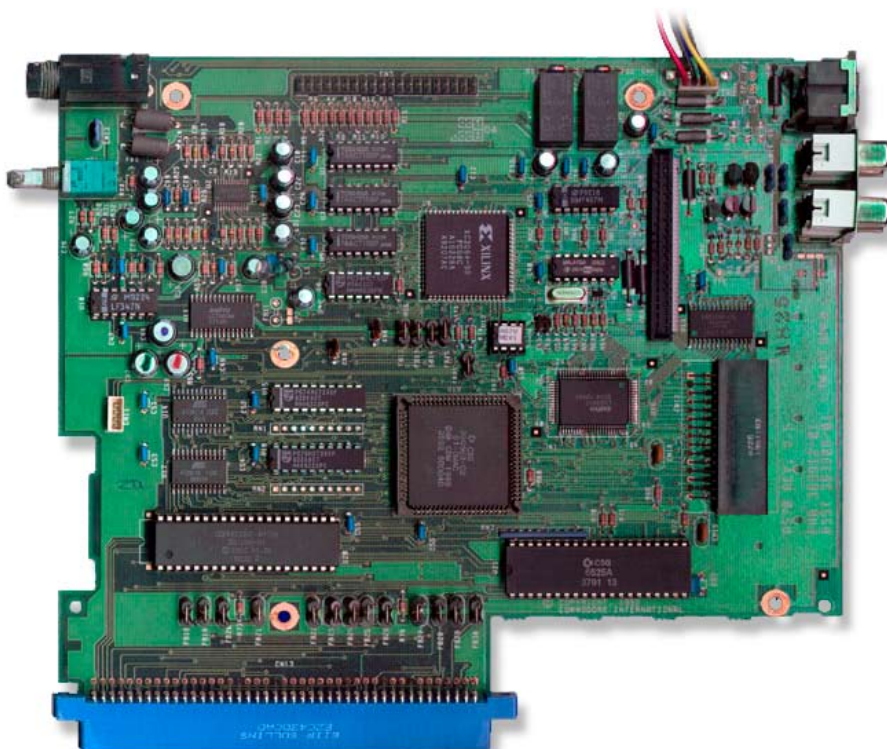
i na tym koniec?

Zapewne większość z Was słyszała o przystawce do A500, dodającej jej bootowalny CD-ROM. Mało tego: zmienia ona naszą Amigę w „prawie” CDTV. Niestety, przystawka ta nie jest przelotowa. Podpięcie jej do komputera całkowicie zamyka dalszą jego rozbudowę poprzez side expansion port. Czy aby na pewno?

Tak, gniazdo zostaje zamknięte. Ale możliwość dodania dodatkowego urządzenia SCSI lub dodatkowej pamięci nadal istnieje. W jaki sposób? Tym osobom, które odważyły się zdjąć obudowę z A570 na pewno rzuciły się w oczy dwa dodatkowe gniazda. Jedno 40-pinowe, umieszczone obok napędu, oraz drugie, 30-pinowe, umieszczone z tyłu urządzenia. Pierwsze służy do zainstalowania dodatkowej pamięci. To z tyłu służy zaś do zainstalowania kontrolera SCSI. Tak więc dziś zajmę się opisem dodatków do A570.

Zacznijmy od dodatkowej pamięci. Dzięki temu gniazdu można dodać dodatkowe 2MB RAM pamięci FAST. Znalazłem tylko urządzenie firmy HK-Computer Vector. Na płycie umieszczone są tylko kostki pamięci oraz zworka, służąca do wyłączania rozszerzenia. Niestety, nie jest ono już dostępne na naszym rynku. Jednak nawet ten fakt nie przekreśla marzeń posiadaczy A570 o dodatkowych 2MB. Z pomocą przychodzi Pan Nicolas Welte z Niemiec. Odkrył on, że przystawka posiada już wbudowany kontroler pamięci. Jedyne, czego brakuje to właśnie kości pamięci. Po tym odkryciu Pan Nicolas wytrawił własną płytkę, wlutował w nią kilka 1MBit-owych DRAMów i w taki sposób stał się posiadaczem rozszerzenia 2MB własnej konstrukcji. Jeśli ta „samoróbka” spotka się ze sporym zainteresowaniem Czytelników – redakcja postara się zdobyć i opublikować schemat karty pamięci.

Następnym dodatkiem poszerzającym możliwości A570 są kontrolery SCSI. Tutaj mamy dwa urządzenia: SCSI-TV/570 firmy AmiTriX oraz Falcon 570 tej samej firmy, która wyprodukowała dodatkowe 2MB FAST.



SCSI-TV/570 jest „dodatkowym pudełkiem” wpinanym z tyłu A570. Na tylnym panelu umieszczone są wyłącznik urządzenia oraz gniazdo DB25, służące do podpięcia dodatkowego napędu. W środku jest miejsce na włożenie dysku 2,5 cala. Producent rekomenduje używanie dysków 2,5 cala z interfejsem SCSI -1. Wspomina również, że niezbyt szybkie dyski standardu SCSI II również powinny działać. Możliwe jest podpięcie dodatkowych dysków oraz streamerów. Niestety nie jest możliwe podłączenie dodatkowych CD-ROM-ów. Przepustowość urządzenia wynosi około 1MB/s. Urządzenie posiada autoboot, działający nawet z systemem 1.3. Przejmując kontrolę nad przystawką, tak więc aby wystartować z CD-ROM-u trzeba je wyłączyć. Po podłączeniu SCSI-TV/570 urządzenia startują w następującej kolejności: HDD, CD-ROM A570, FDD. Dyski podłączone do urządzenia można partycjonować programem HDToolBox.

Falcon 570 jest kartą złożoną z dwóch płytek, podłączonych pod kątem prostym oraz wpinanych w tył A570. Posiada ona tylko wyprowadzenie na gniazdo SCSI DB25. Niestety brak dokumentacji uniemożliwia charakterystykę tego urządzenia. Na płycie umieszczone są dwie zworki. Można się tylko domyślać, że służą one do włączania/wyłączania rozszerzenia. Dyski podpięte do tego urządzenia również partycjonuje się programem HDToolBox. Prawdopodobnie urządzenie działa z systemem 1.3 oraz posiada autoboot.

Reasumując: dokupienie do Amigi 500 napędu A570 nie ogranicza się tylko do dodatkowego startującego CD0, ale stwarza również możliwość podpięcia kilku dysków twardych i rozszerzenia pamięci.

Milek



Sprawy techniczne i *MegaRamHD*

Sporo świeżo upieczonych użytkowników tego rozszerzenia Elsat-u do A500 ma kłopoty z jego uruchomieniem. Z tego powodu postanowiłem dziś opisać, jak powinna wyglądać konfiguracja i instalacja pamięci / dysku w tym urządzeniu.

Pamięć

MegaRamHD obsługuje TYLKO SIMM-y 1 MB lub 4 MB. SIMM-y muszą być 30-pinowe (jednostronne). Przy instalacji pamięci trzeba się zdecydować na jeden ich rodzaj SIMM-ów. Tutaj są dostępne tylko 3 rodzaje konfiguracji:

- 2 MB (2 SIMM-y po 1 MB)
- 4 MB (4 SIMM-y po 1 MB)
- 8 MB (2 SIMM-y po 4 MB)

Urządzenie nie jest autokonfigurowalne, więc o ilości zainstalowanej pamięci trzeba je poinformować ustawieniem zworek. Służą do tego zworki JP1 i JP2. Każdą z nich można ustawić w pozycji 1-2 lub 2-3. Oto ustawienia zworek odpowiednie dla ilości posiadanej pamięci:

- 0 MB – JP2 na 1-2; JP1 na 1-2
- 2 MB – JP2 na 2-3; JP1 na 2-3
- 4 MB – JP2 na 2-3; JP1 na 1-2
- 8 MB – JP2 na 1-2; JP1 na 2-3

Oprócz tego na płycie znajdziemy dwie inne zworki. Jeśli zdejmujemy jedną z „SHUTUP RAM” - wyłączymy całkowicie kontroler pamięci. Jeśli zewrzymy „SHUTUP HD” - wyłączymy kontroler HDD. W przypadku tych zworek aż się prosi, aby dzięki nim wyprowadzić przełączniki na zewnątrz obudowy.

Dysk twardy

Elsat MegaRamHD widzi tylko dyski standardu AT-BUS, lecz i przy nich jest bardzo

„wybredny”. Nie każdy dysk zostanie wykryty, więc przy wkładaniu „nowego” trzeba liczyć na odrobinę szczęścia. Urządzenie posiada auto-boot, tak więc nawet w systemie 1.3 jest możliwy start z dysku twardego. Sterownik urządzenia jest zapisany w ROM-ie MegaRamHD. Nie trzeba go kopiować na dysk systemowy. Jak sprawdzić, czy przystawka wykryła nowo podłączony dysk? Za pomocą programu At-Bus_Info. Jeśli urządzenie wykryło dysk – program nas o tym powiadomi. Jeśli dysk został poprawnie wykryty – partycjonujemy go za pomocą następnego programu z firmowej dyskietki - HardDiskPrep. Partycje dysku potraktowane tym programem nie będą widoczne w innych Amigach (np. przy podłączeniu do A600 lub A1200). Po tych zabiegach trzeba tylko sformatować partycje i już można się cieszyć z używania twardego dysku w naszej A500. Programy At-Bus_Info i HardDiskPrep są dostarczone na oryginalnej dyskietce, dostarczanej razem z kontrolerem MegaRamHD.

Tuning

W C&A Fan Nr 2 pisałem, że nie jest możliwe podpięcie dwóch urządzeń do taśmy sygnałowej kontrolera (np. dwóch dysków lub dysku i CD-ROM-u) bez odpowiednich przeróbek. Owe przeróbki są bardzo proste. Posiadając odpowiednie programy oraz starą taśmę do podłączania dwóch urządzeń IDE, można je wykonać w parę minut. Najpierw musimy zaktualizować ROM kontrolera, tak aby wykrywał dwa urządzenia. Bez obawy, wcale nie będzie trzeba rozkręcać kontrolera i wyciągać z niego układów. Z pomocą przyjdzie nam tu program autorstwa Tomasza Pęczka o nazwie MegaRamFix. Program jest dostępny na Aminecie (megaramfix.lha). Po jego uruchomieniu ROM urządzenia Elsatu zostanie zaktualizowany. Autor zaleca, aby program został

uruchomiony na samym początku sekwencji startowej. Można teraz przystąpić do następnej części przeróbki.

Oczywiście nie muszę przypominać o przestawieniu zworki drugiego dysku lub CD-ROM-u na opcję SLAVE. Wyłączamy komputer, z MegaRamHD wypinamy oryginalną taśmę i wpinamy nową, służącą do podpięcia dwóch urządzeń. Na wszelki wypadek dodam, że musi być to taśma starego typu. Do niej podpinamy stary dysk (do złącza na samym końcu taśmy) oraz drugi napęd (do złącza w środku taśmy). Teraz możemy przejść do ostatniego etapu. Musimy powiadomić system o istnieniu nowego napędu. Z pomocą przyjdzie nam sterownik ide.device, autorstwa Pawła Stypiuli, również dostępny na Aminecie (ide.lha). Opis jak go zainstalować jest umieszczony w archiwum. Jeśli drugim napędem jest dysk np. z A600 lub A1200 – partycje zostaną utworzone automatycznie. Jeśli dysk jest „pusty” - należy go potraktować systemowym programem do obsługi dysków, czyli HdToolBox. Jeśli drugim napędem jest CD-ROM, należy jeszcze zainstalować system plików CD.

Osobiście zainstalowałem AmiCDFS i wszystko działa mi bez zarzutu. Jeśli nie wiecie skąd wziąć paczkę z tym systemem plików, powinniście się już domyślić: z Aminetu (amicdfs240.lha). Na zakończenie dodam, że odczyt z CD-ROM-u jest o 1/4 wolniejszy, niż z pierwszego dysku. Testy odczytu z programu SysInfo wykazywały około 275 kB/s. Lecz mimo tej niedogodności, możliwość korzystania z CD-ROM-u na Amidzie 500 jest niezastąpiona. Aby ułatwić życie naszym Czytelnikom, na dyskietce dołączonej do tego numeru znajdziecie wszystkie programy niezbędne do uruchomienia zestawu MegaRamHD + CD.

Milek

Amiga Guide

szkółka



Każdy programista, grafik, czy muzyk, po zakończeniu swojej pracy chce przedstawić swoje dzieło, tworząc do niego instrukcję, opis funkcji, czy też historię powstania. W zależności od platformy, dominują różnego rodzaju formaty; od danych tekstowych, poprzez pdf'y, na html'u kończąc.

Amigowcy przez dłuższy czas korzystali (i nadal korzystają) z tego pierwszego, zamieszczając dokumentację w pliku tekstowym, najczęściej z rozszerzeniem „.readme”. Pojawienia się „Amiga Guide”, zmieniło formę przekazywanych treści. A jego najczęstsze wybieranie jako formatu dla dokumentacji, to dowód niesłabnącej popularności. Wystarczy spojrzeć na Aminet.

No tak, ale czym zatem jest „Amiga Guide”. Jest to rodzaj hipertekstu, w którym oprócz samych danych tekstowych, występują także odnośniki (linki) do następnych danych. Linki mogą tworzyć spisy treści, indeksy, a także mogą występować jako pojedynczy wyraz w zdaniu. Poza tym, zaimplementowane komendy pozwalają na różnego rodzaju uatrakcyjnienie samego tekstu, jak i uruchamianie skryptów Arexxa, a dzięki obsłudze poleceń AmigaDosu, możliwe jest wyświetlenie każdego rodzaju danych graficznych, muzycznych, czy animacji, włączając w to uruchamianie programów.

Swoj opis oparę na dwóch, najczęściej stosowanych wersjach; wersji 39, która pojawiła się wraz z AmigaOS 3.0, oraz na wersji 40, dostępnej od systemu 3.1.

Wszelkie komendy w hipertekście rozpoczynamy od postawienia znaku „@”. Przeglądarka wie wtedy, że po małym będzie komenda, a nie zwykły tekst, do wyświetlenia.

Komendy można zatem podzielić na ogólne (globalne), komendy mające zastosowanie w poszczególnych rozdziałach (węzłowe), atrybuty, które można wstawiać w dowolnym miejscu w hipertekście oraz komendy odnoszące się bezpośrednio do wyświetlanego tekstu.

Komendy globalne – grupa komend, które narzucają pewne rozwiązania odnoszące się do całości skryptu AG:

- **@DATABASE** (nazwa dokumentu) – rozpoczyna każdy plik AG, nadając mu nazwę. Przykład: @DATABASE Kurs Amiga Guide.

- **@\$VER** (wersja) – dzięki tej komendzie, nadajemy numer wersji naszego pliku. Warto dodać, że będzie ona także wyświetlana przez systemowe polecenie VERSION. Przykład: @\$VER Wersja 1.0, Luty 2010.

- **@AUTHOR** (autor) – tutaj podajemy dane autora, komenda nie jest wymagana. Przykład: @AUTHOR Don Rafito.

- **@(C)** (prawa autorskie) – wiadomo, tutaj wpisujemy, do kogo należą prawa autorskie, spisywanego w formacie AG tekstu. Komenda nie jest wymagana. Przykład: @(C) FanCA.

- **@FONT** (nazwa czcionki) (rozmiar czcionki) – za pomocą tej komendy, narzucamy rodzaj i wielkość czcionki, zastosowanej w całym dokumencie. Komenda, nie jest wymagana i jeśli z niej nie skorzystamy, przeglądarka skorzysta z systemowo ustawionego fontu. Przykład: @FONT topazpl.font 8.

- **@MASTER** (nazwa katalogu) – dzięki tej komendzie, określamy katalog, w którym znajduje się plik tekstowy, z którego dane posłużyły nam do stworzenia dokumentacji w AG. Nie jest ona wymagana, a obecnie prawie w ogóle nie stosowana.

- **@HEIGHT** (liczba wierszy) – określa maksymalną liczbę wierszy w rozdziałach. Komenda nie jest wymagana.

- **@WIDTH** (liczba znaków) – analogicznie do poprzedniego, maksymalna liczba znaków w wierszach. Nie wymagana.

- **@HELP** (ścieżka/węzeł) – określa ścieżkę dostępu dla rozdziału (węzła), który zostanie wyświetlony po naciśnięciu HELP w przeglądarce (np. w MULTIVIEW). Nie jest ona wymagana i przy jej niezastosowaniu gdy wywołamy ową pomoc, wyświetli się nam instrukcja użytkownika przeglądarki.

- **@INDEKS** (ścieżka/węzeł) – tutaj określamy dostęp rozdziału, będący indeksem (np. alfabetycznym) i który wyświetli się nam po naciśnięciu przycisku INDEKS w przeglądarce. Nie wymagana.

- **@MAKRO** (nazwa) (komendy) – ta dość skomplikowana komenda, wprowadzona od wersji 40, daje nam możliwość odgórnego wymuszania pewnych zachowań na tekście, takich jak pogrubienie, czy kursywa. Np. wprowadzenie: @MAKRO kursywa „@{I}\$1@{UI}” daje możliwość wymuszania pochylenia tekstu (tutaj określonego jako parametr \$1), w dowolnym momencie dokumentu, poprzez wpisanie @ {kursywa „pochylony tekst”}, gdzie „pochylony tekst” może być dowolnym ciągiem wyrazów, czy całych zdań. Komenda nie jest wymagana.

- **@ONOPEN** (komenda Arexxa) – pozwala na odpalenie skryptu Arexxa, w momencie otwierania pliku AG. W przypadku błędów w skrypcie, dokument nie zostanie wyświetlony. Od wersji 40. Nie wymagana.

- **@ONCLOSE** (komenda Arexxa) – podobnie to tej powyżej, ale z tą różnicą, że skrypt uruchomi nam się w przypadku zamykania przeglądane dokumentu. Także od wersji 40. Komenda nie wymagana.

- **@REM** (komentarz), oraz **@REMARK** (komentarz) – umożliwia wstawianie komentarzy w skrypcie AG, które nie będą widoczne w przeglądarce. Obie nie wymagane.

- **@TAB** (liczba) – tutaj definiujemy rozmiar tabulatora. Domyślnie składa się z ośmiu znaków. Od wersji 40 i nie wymagana.

- **@WORDWRAP** – dzięki tej komendzie wyjustujemy wszystkie paragrafy. W wersji 39, usuwa ona także widoczne znaki końca linii (enter). Nie wymagana.

- **@SMARTWRAP** – występująca od wersji 40, nieco ulepszona wersja poprzedniej komendy. Nie wymagana.

- **@NODE** (nazwa) (tytuł) – tym poleceniem otwieramy dany rozdział (węzeł), podając jego nazwę, która będzie się później odnosić do linków, oraz tytuł, który wyświetli się nam na górnej belce okna przeglądarki. Warto dodać, że pierwszy rozdział w naszym dokumencie (będący wstępem, spisem treści, itp.) musi nosić nazwę MAIN. Dalsze następne są już dowolne. Przykład: @NODE MAIN „Kurs Amiga Guide - spis treści”, lub dalszy @NODE tekst „Przykłady pracy z tekstem”, gdzie wyraz tekst, posłuży w kolejnych komendach jako link do tego właśnie węzła (rozdziału).

- @ENDNODE – zakańcza dany węzeł (rozdział). Obie komendy, są niezbędne do prawidłowo napisanego skryptu. Jeśli będą w nim błędy, czy też literówki, dokument AG w ogóle się nam nie wyświetli.

Komendy węzłowe – stosuje się je dla poszczególnych rozdziałów, charakteryzując np. wygląd danego węzła, który będzie się różnił od innych. Komendy te muszą znajdować się w danym rozdziale (po @NODE i przed @ENDNODE):

- @TITLE (tytuł) – tutaj wstawiamy tytuł naszego rozdziału, który wyświetli się nam na górnej belce przeglądarki. Warto dodać, że jest ona niejako zamiennikiem komendy @NODE (nazwa) (tytuł), ale tylko w części dotyczącej nadania tytułu dla danego węzła. Komenda ta, nie działa tak jak ta poprzednia, więc nie otwiera nam rozdziału.

- \ - **backslash poprzedzający znak @**, umożliwia pokazanie go w tekście (przeglądarka wie, że w tym przypadku mała jest tekstem a nie początkiem komendy), ale jeśli chcemy aby backslash także był widoczny w dokumencie, również go poprzedzamy tą komendą (\).

- @TOC (ścieżka/nazwa) – domyślnie spisem treści jest rozdział MAIN. Dzięki temu poleceniu, sami wskażemy, który węzeł ma być wyświetlony po naciśnięciu SPIS TREŚCI w przeglądarce.

- @NEXT (ścieżka/węzeł), @PREV (ścieżka/węzeł) – tutaj definiujemy, który rozdział ma być następny (po kliknięciu na NASTĘPNY w przeglądarce), a który ma być poprzedni (POPRZEDNI w przeglądarce). Przykład:

@NEXT kolor

@PREV MAIN.

Pewnym dość ciekawym rozwiązaniem jest to, że część komend globalnych można stosować w danym rozdziale. Będą one wtedy odnosić się tylko do danego węzła. Warunek, jak poprzednio, muszą znaleźć się między komendą @NODE a @ENDNODE. To tych komend zaliczamy: @TAB, @FONT, @HELP, @INDEX, @MACRO, @ONOPEN, @ONCLOSE, @WORDWRAP, @SMARTWRAP.

Kolejna grupa to **atrybuty**. Stosowanie ich możliwe jest bezpośrednio w tekście. Dzięki temu, można zdefiniować poszczególne, wyrazy, całe zdania, czy nawet całe akapity, jako linki do innych rozdziałów, poleceń, czy komend zewnętrznych (Arexx, AmigaDos). Formuła atrybutu składa się z: @{(nazwa) (komenda)}, gdzie w nazwa wpisujemy owy tekst (wyraz, zdanie, akapit), a w komenda wpisujemy, jedno z niżej wymienionych poleceń:

- **LINK** (ścieżka/nazwa) (numer linii) – to polecenie występuje najczęściej, dzięki niemu definiujemy dany wyraz, czy zdanie jako link do danego rozdziału. W numer linii wpisuje-

my od której linijki ma być wyświetlony dany węzeł. Domyślnie jest on pierwszy (jeśli nie wstawimy tam nic). Przykład: @{„Operacje na kolorach” LINK kolor}.

- **ALINK** (ścieżka/nazwa) (numer linii) – podobnie jak wyżej, tylko z tą różnicą, że dany rozdział będzie wyświetlony w nowym oknie. Działa tylko do wersji 39.

- **CLOSE** – zamyka nam okno z otwartym, za pomocą poprzedniej komendy, węzłem.

- **RX** (skrypt Arexxa) – zdefiniowany wyraz (link) za pomocą tej komendy uruchamia wpisany skrypt Arexxa.

- **RXS** (komenda) – naciśnięcie takiego linku wykonuje Arexxowy „string file”.

- **SYSTEM** (komenda AmigaDosu) – dzięki temu poleceniu nasz link wykonywać będzie wiersz poleceń (wpisany w komenda AmigaDosu), identycznie jak w przypadku SHELLA. Dzięki temu możemy uruchomić program, wyświetlić grafikę, czy inny rodzaj danych. Ważne jest, aby podawać pełną ścieżkę do pliku w przypadku gdy nie znajduje się on w tym samym katalogu co nasz dokument AG. Przykład: @{„klikając na ten klawisz” SYSTEM „multiview CA.iff „} – tutaj nasz wskazany obrazek szukany jest przez MULTIVIEW w tym samym katalogu co plik dokumentu.

- **QUIT** – wskazany link dzięki tej komendzie zamknie nam okno przeglądarki. Polecenie to jest dość stare, gdyż nie działa od systemu 3.0 (wersji 39).

Ostatnia grupa, to **komendy które odnoszą się bezpośrednio do napisanego tekstu**. Dzięki nim, możemy justować dane akapity, pogrubiać litery, czy kolorować wyrazy i zdania:

- @B – pogrubianie tekstu po komendzie. Od wersji 39 (podobnie jak pięć następnych).

- @UB – wyłączanie pogrubiania.

- @I – włączenie pochyleń (kursywy).

- @UI – wyłączanie kursywy.

- @U – od tej komendy nasz tekst będzie podkreślony.

- @UU – wyłączenie podkreślenia.

- @PLAIN – polecenie to wyłącza wszystkie powyżej. Od wersji 40.

- @APEN (numer koloru w paletce Workbench) – korzystając z palety systemu (w zależności od głębi ekranu, od koloru „0” do koloru o numerze „255”), możemy zmienić kolor wyświetlanego tekstu.

- @BPEN (numer koloru w paletce Workbench) – to co wyżej, tylko w tym przypadku zmieniamy kolor pod tekstem.

- @LINDENT (liczba) – tym poleceniem określamy liczbę spacji wstawianych na początek akapitu.

- @PARD – odwołuje wszystkie cztery, wyżej wymienione polecenia, przywracając standardowy kolor tekstu i tła, oraz standardową czcionkę. Komendy te występują od wersji 40.

Warto dodać, że powyższe polecenia, począwszy od pogrubienia tekstu, na zmianie koloru tła skończywszy, można ze sobą łączyć (wstawiając jedna po drugiej), żeby np. otrzymać tekst, który będzie pochylony, pogrubiony, o zabarwieniu białym i na dodatek na niebieskim tle. Przykład:

(...) @BPEN „2”@I nadzieję @PLAIN @PARD

(...) @APEN „3”@B przykład @PLAIN @PARD (...).

- @JCENTER – centrowanie tekstu.

- @JLEFT – wyjustowanie tekstu do lewej.

- @JRIGHT – wyjustowanie do prawej.

- @CODE – wyłącza wszystkie trzy powyższe, przywracając justowanie domyślne. Wszystkie występują od wersji 40.

- @TAB – po wstawieniu tej komendy, na wyjściu będą widoczne znaki tabulatora, zamiast spacji. Wszystkie wersje. Pozostałe, wymienione poniżej, występują już tylko od wersji 40.

- @BODY – przywraca domyślny format tekstu.

- @SETTABS (liczba)...(liczba) – ustawia serię tabulatorów o kolejne liczby spacji.

- @CLEARTABS – odwołuje poprzedni rozkaz, ustawiając wartości domyślne dla tabulatorów.

- @LINE – powoduje wstawienie w dokumencie pustej linii, bez rozpoczynania paragrafu.

- @PAR – używany w rozdziale z włączonym @SMARTWRAP, powoduje wstawienie dwóch pustych linii.

- @AMIGAGUIDE – powoduje wstawienie w tekście pogrubionego napisu „Amiga Guide”.

Pisanie dokumentacji w Amiga Guide, jest dosyć ciekawym sposobem na zaprezentowanie treści. Znam przykłady skryptów (nie tylko instrukcji do oprogramowania), które przedstawiają w tej postaci opowiadania, przepisy kulinarne, spis dowcipów, a nawet całych fanzinów, zawierających także dane graficzne, czy dźwiękowe. Możliwości są naprawdę spore. Cały sęk zależy od naszej wyobraźni i chęci. Choć, gdy tych ostatnich troszkę brakuje, można sobie ułatwić sprawę i skorzystać z programów dostępnych w sieci (m. in. wspomniany już nie raz Aminet), które po wprowadzeniu danych same wygenerują gotowy plik z rozszerzeniem „guide”. Niemniej jednak warto wiedzieć, jak to wygląda od kuchni. Zapraszam zatem, do eksperymentów z hipertekstem, a na ułatwienie proponuję przykład skryptu, składający się z podstawowych komend, dołączonego do obecnego numeru C&A Fan.

Don Rafito

LHA cz. 2

Po długiej nieobecności zapraszam do drugiej części kursu Lha. Dzisiaj napiszę coś o zmiennej oraz o wzorcach plików i katalogów. Informacje z tej części przydadzą się nie tylko użytkownikom Lha lecz również wszystkim którzy muszą czasami skorzystać z AmigaDOS'u.

Zacznijmy może od zmiennych które obsługuje Lha. A obsługuje, jak zresztą większość Amigowego softu, dwa rodzaje zmiennych: globalne i lokalne. W momencie uruchomienia, program przyjmuje wartości zapisane w zmiennej o nazwie **LHAOPTS**.

Wartość zmiennej jest interpretowana dosłownie, czyli tak jak została ustawiona, i łańcuch opcji jest „wstawiany” zaraz za poleceniem lha. Zaraz na przykładzie wytłumaczę jak to działa. Najpierw jednak napiszę jak się przed tym ustrzec. Być może kiedyś, komuś się to do czegoś przyda. Tak więc jeżeli z jakichś powodów chcemy aby Lha nie odczytywał tej zmiennej należy podać parametr – **I** (duże I jak Iza). Czas na przykład. Jeżeli użyjemy polecenia:

```
Setenv LHAOPTS -N -b128
```

Lha nie będzie wyświetlać postępów w rozpakowywaniu archiwum oraz będzie używać 128KB bufora dla wykonywanych operacji.

Oczywiście nie będzie to trwać wiecznie. A ściślej mówiąc będzie tak dopóki nie dokonamy resetu Amisi lub po prostu zmienimy parametry zmiennej globalnej LHAOPTS.

Ala co zrobić jeżeli chcemy aby zawsze lha korzystał z wybranych przez nas parametrów. Normalnie musielibyśmy za każdym razem ustawiać tą zmienną, ale jest pewien sposób. Można skorzystać ze zmiennej ENVARC : LHAOPTS. Przykładowo można wydać komendę:

```
setenv ENVARC:LHAOPTS -b64
```

Komenda powoduje że zmienna LHAOPTS przyjmuje wartość -b64 zawsze po resecie komputera.

Na podobnej zasadzie działa wiele programów nie tylko tych odpalanych z poziomu AmigaDOSu lecz również z Workbenchu za pomocą ikony.

Najczęściej archiwum składa się z kilku do kilkudziesięciu plików. Sytuacja gdy pakujemy jeden plik należy do rzadkości. Aby nie potrzeba było wpisywać wszystkich plików po kolei wymyślono system znaków globalnych. W skrócie chodzi o to, aby znaleźć cechy wspólne plików, a różnice zastąpić jakimś symbolem. Lha korzysta w tym zakresie ze standardowego zestawu znaków globalnych AmigaDOSu. Dzięki temu rozwiązaniu można budować wzorce nazw zarówno plików jak i katalogów.

Przejdę teraz do omówienia poszczególnych sposobów oznaczania większej ilości obiektów. Aby nie było niejasności postaram się opis okraszyć jasnymi przykładami.

Pierwszym znakiem jakim się zajmę jest pytajnik (?). Symbolizuje on jakikolwiek pojedynczy znak. Na przykład:

a? – spakuje wszystkie pliki dwuliterowe zaczynające się na „a”. Czyli pliki ab, az, a9 zostaną dodane do archiwum.

ab?d – spakuje wszystkie czteroliterowe pliki zaczynające się na „ab”, a kończące na literę „d”. Spakowane zostaną pliki abcd, ab9d, ab_d, ale już plik o nazwie „abd” zostanie pominięty.

Oczywiście w naszym wzorcu plików możemy wstawiać wiele pytajników. Nie jesteśmy



ograniczeni tylko do jednego. Na przykład: `a??z?` – weźmie pod uwagę wszystkie obiekty pięcioliterowe o nazwie zaczynającej się na „a”, po niej mogą wystąpić dwa dowolne znaki, a później musi być „z” i jeden dowolny znak.

Kolejnym takim znakiem jest gwiazdka (*). Zastępuje ona dowolne znaki w dowolnej ilości (w tym ciągi o długości 0 (zero)). Na przykład:

* – często stosowana dla oznaczenia wszystkich obiektów w danym miejscu (katalogu).

`a*` – zastępuje dowolną nazwę o dowolnej długości zaczynającą się na „a”

`a*z` – zastępuje dowolną nazwę o dowolnej długości zaczynającą się na „a”, a kończącą na „z”.

`a*b*c` – określa wszystkie nazwy zaczynające się na „a”, potem może wystąpić ciąg dowolnej długości (proszę pamiętać że może on przybrać wartość 0 co oznacza że plik „abc” także zostanie wybrany) aż do litery „b” która wystąpić musi i tak samo do litery „c”, która jest ostatnią literą nazwy.

*.lha – pozwoli na przeprowadzenie operacji na wszystkich archiwach lha.

Kolejnym znakiem globalnym, z którego mogą korzystać użytkownicy lha jest hash czyli krzyżyk (#). Oznacza on występowanie danego znaku dowolną ilość razy. Na przykład ciąg „#?” jest równoważny „*” gwiazdce. Oznacza dowolną ilość pytańników, które określają dowolny jeden znak.

#a – określa nazwy składające się z dowolnej ilości litery „a” i tylko jej. Np.: a, aa, aaaaaa.

a#bc – weźmie pod uwagę wszystkie nazwy zaczynające się na „a” i dowolną ilość (w tym 0) liter „b”, a kończące się na „c”, czyli będą to pliki takie jak: abc, ac, abbbbbbcb.

#(ab)#(cde)fg –weźmie pod uwagę wszystkie pliki w których występuje, lub nie, fraza „ab” raz ,lub wiele razy, to samo tyczy się „cd”. Nazwa obiektów musi kończyć się na „efg”. Będą to takie pliki i katalogi jak: „abababdcdefg”, „abefg”, „cdefg”, „efg”, „abcdefg”.

Do określania wzorca plików można używać także nawiasów kwadratowych. Zwróć w nim trzeba zakres znaków o jaki nam chodzi, rozdzielony kreską (-) minusem. Np. [1-9], [a-z], [abcd]. W tym ostatnim przypadku wzięte pod uwagę będą jedynie litery w nawiasie czyli „a”, „b”, „c”, „d”. Można także zdefiniować kilka zakresów, ale wtedy trzeba je rozdzielić przecinkiem, np.: [a-d,g-m].

Oczywiście znaki globalne działają jak poprzednio, jednak biorą pod uwagę dowolny znak lub znaki (zależnie od użytego wcześniej znaku globalnego) ale tylko z podanego zakresu. Można powiedzieć że wartości w nawiasach kwadratowych stanowią pewnego rodzaju ograniczenie dla innych znaków globalnych.

Zaraz wszystko rozjaśnię odpowiednimi przykładami. `mod.[1-5]` – określi wszystkie pliki 5 znakowe, rozpoczynające się od słownych liter „mod”, po których musi nastąpić kropka. Po niej musi znajdować się jedna cyfra z przedziału od 1 do 5.

Często osoby piszące programy oznaczają kolejne etapy swojego dzieła numerem wersji. Pokażę teraz jak oznaczyć do spakowania wybrane wersje programu.

`ver_[1-3].[0-9].[a-z]` – określa każdy 9 znakowy plik rozpoczynający się od „ver_”, po którym znajduje się cyfra „1”, „2” lub „3”. Następnie kropka i dowolna cyfra (jedna) z zakresu od 0 do 9, potem kropka i jedna dowolna litera z zakresu od a do z. Prawda że praktyczne, gdy właśnie powstała wersja 4 danego programu i chcemy umieścić w jednym archiwum wszystkie poprzednie? O ile oczywiście konsekwentnie stosowaliśmy ten schemat nazewnictwa. Może jeszcze kilka przykładów jakie pliki zaznaczy ten schemat, będą to „ver_2.1.c” lub „ver_3.9.z”

#[a-z 0-9] – dokona operacji na każdym pliku zawierającym dowolną ilość znaków i cyfr. Czyli np. „plik”, „plik001”, „Amiga500”. Polecenie nie obejmie nazw takich jak „plik.001”, dlatego że znajduje się w nim kropka nie określona w schemacie.

*.[abcd] – określa wszystkie pliki które po „.” (kropce) mają znak „a”, „b”, „c” lub „d”. Przydatne przy kodach źródłowych programów.

Kolejnym elementem, który może być używany do określania plików i katalogów są nawiasy okrągłe (). Pomiedzy nimi należy zdefiniować listę wartości w postaci ciągu, którą jesteśmy zainteresowani. Są szczególnie użyteczne razem z wcześniej poznanymi znakami globalnymi. Ich zastosowanie i działanie najlepiej poznać na przykładach.

*.(guide|iff|c) – przeprowadzi operację na wszystkich plikach o rozszerzeniu „guide”, „iff” lub „c”.

(abc|cde|fgh) – weźmie pod uwagę jedynie pliki o nazwach „abc”, „cde” oraz „fgh”.

(*a|*b|*.doc|pic*) – wybrane polecenie wykona operację na plikach o rozszerzeniu „a”, „b”, „doc” oraz na plikach, których nazwa rozpoczyna się na „pic”. Nawiasy mogą być oczywiście zagnieżdżone. Ciąg (abc|cde|efg) jest równoważny ciągowi abc cde efg.

Kolejnym użytecznym znakiem globalnym jest tylda (~). Powoduje ona wykluczenie danego ciągu tekstowego. Jeżeli nazwa pliku lub katalogu zawierać będzie ten ciąg, to żądana operacja nie zostanie na niej wykonana. Wygląda to mniej więcej tak.

~(file) – weźmie pod uwagę wszystkie pliki, oprócz tych z nazwą „file”. Tu mała uwaga. To nie jest to samo wyrażenie co ~file.

Tylda neguje tylko wyrażenie bezpośred-

nio występujące po niej, a nie jak niektórzy sądzą wszystko co po niej napisano.

(~a?) – wyłączy z wyboru wszystkie dwuliterowe nazwy zaczynające się na „a”, czyli takie jak „a1”, „az”, itp. Wszystkie pozostałe dwuliterowe nazwy zostaną wzięte pod uwagę.

~(a?) – wykona żadaną operację na wszystkich plikach oprócz dwuliterowych zaczynających się na „a”. Na pliku o nazwie „abb” zostanie wykonana operacja tak samo jak na pliku „cd”, ale już obiekt o nazwie „ab” lub „a5” zostanie pominięty. Jako ciekawostkę podam ciąg, który spowoduje, że operacja będzie wykonana na pustym zbiorze plików. Robię to, tylko dla lepszego zrozumienia. W tym wypadku schemat wygląda tak ~(#?). Tylda neguje wyrażenie, które opisuje wszystkie nazwy.

~lha – bierze pod uwagę wszystkie pliki które nie zaczynają się na „l” i nie kończą na „ha”. Pliki typu „uha” lub „abc_lha” zostaną dołączone do zbioru, natomiast pliki o nazwach takich jak „labcdha” czy „lha” już nie.

Kolejnym znakiem jest procent (%). Reprezentuje on pusty ciąg znakowy. Znajduje on zastosowanie jedynie w wyrażeniach w nawiasach. Co rozumiałe nie może on być łączony ze znakiem #. Wyrażenie #% nie ma tak naprawdę sensu ponieważ % oznacza zawsze zero znaków. Na przykład:

lha(.doc|.iff|%)

– wykona operację na plikach „lha.doc”, „lha.iff” oraz „lha”.

Ostatnim znakiem specjalnym omówionym w artykule jest apostrof ('). Powoduje anulowanie znaczenia znaku specjalnego, który po nim następuje. Tak to prawda na Amidze nic nie stoi na przeszkodzie używanie znaków takich jak * czy ? w nazwie plików i katalogów. Spróbujcie zrobić coś takiego na PeCecie. Należy go używać wyłącznie ze znakami ?*#[()~-%' (ten ostatni to właśnie apostrof). Na przykład: hello'? – wykona operację wyłącznie na pliku o nazwie „hello”

lha'** – wykona operację na wszystkich plikach, które zaczynają się od liter „lha”
a'b'c – będzie się odnosić do pliku „a'bc” gdyż dwa apostrofy razem są traktowane jak jeden.

Chciałbym dodać iż lha korzysta z udogodnień i ustawień w Localach. Dzięki temu może również korzystać z wszystkich funkcji biblioteki locale. Czyli różne specjalne znaki narodowe nie stanowią dla niego problemu. Na koniec życzę czytelnikom przeprowadzania wszystkich operacji zawsze na poprawnie wskazanym zbiorze katalogów i plików. Myślę, że dzięki temu artykułowi korzystanie z Amigowego Shella stanie się znacznie łatwiejsze, i to nie tylko z lha.

MrMat

Asystent Pro 2.0

domowy budżet pod kontrolą

Pieniądze (niestety) odgrywają w życiu człowieka bardzo ważną rolę i to od już zarania dziejów. Obecnie nie ma już chyba dziedziny, w której bez kasy nie da się cokolwiek zdziałać. Operacje finansowe państw, firm, przedsiębiorstw, placówek państwowych i prywatnych, a nawet gospodarstw domowych opierają się na umiejętnym (bądź nieumiejętnym) liczeniu przychodów i planowaniu wydatków. Takie czynności noszą właśnie nazwę prowadzenia budżetu.

Dzięki komputerom i aplikacjom biurowym, które notabene zajmują jedną z głównych części powstającego dziś oprogramowania, prowadzenie operacji finansowych jest szybsze, efektywniejsze i dużo łatwiejsze niż kiedyś. Nawet w domach, zwyczajny kalkulator, długopis i kartka papieru, zaczynają być wypierane przez programy komputerowe, liczące bilans dochodów i wydatków, prowadzące statystykę, z rozmieszczaniem jej na wykresach, żywcem wziętych z giełd papierów wartościowych włącznie.

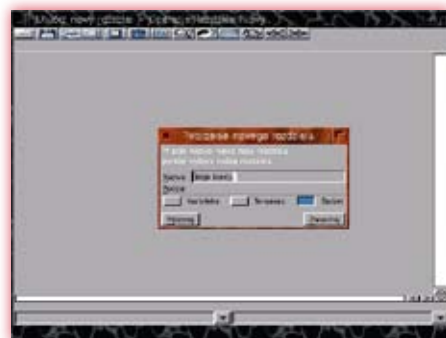
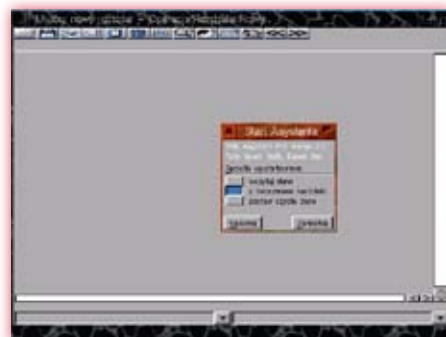
Program, który mam zamiar przedstawić, dzięki zaimplementowaniu aż trzech modułów, można nazwać swego rodzaju, mini kombajnem biurowym. I choć jego możliwości, w połączeniu jeszcze z kilkoma innymi aplikacjami, świetnie nadają się do prowadzenia małej działalności (sklepu, warsztatu, itp.), to można go także wykorzystać do domowych zastosowań. Adresy i numery telefonów rodziny, przyjaciół, spis płyt z filmami i z muzyką, to zadania dla „Kartoteki”. Dzięki „Terminarzowi” nie zapomnimy o urodzinach cioci, weekendowym wyjeździe, czy wizycie u lekarza. A moduł „Budżet” zadba o to by nasze domowe wydatki, rachunki, zakupy, czy stan konta w banku, był pod stałą kontrolą.

Domniemując z tytułu, skupimy się tutaj na opisanu tego trzeciego.

Może kilka słów o samym programie. Producentem jest polska firma Twin Spark Soft. Więc jak się można domyśleć, nie będą mieli z jego obsługą użytkownicy posługujący się tylko naszym ojczystym językiem. Interfejs jest dość prosty, schludnie wykonany i czytelny. Klawisze w oknach stylizowane są na te z pakietu MUI, choć sam „Asystent” z niego nie korzysta. Może to trochę nie odpowiadać niektórym osobom, przyzwyczajonym do programów bardziej ubarwionych, biorąc pod uwagę fakt, że poza klawiszami, reszta kolorystyki jest raczej szarawa, a rysowanie wykresów odbywa się tylko za pomocą kilku kolorów, to osobiście uważam że programy tego typu, mają być przede wszystkim funkcjonalne, a nie świecić jak choinka w Boże Narodzenie. Poza tym fakt niewielkich wymagań sprzętowych, umożliwia nawet posiadaczom Amigi 500 poprowadzenie swojego domowego biura.

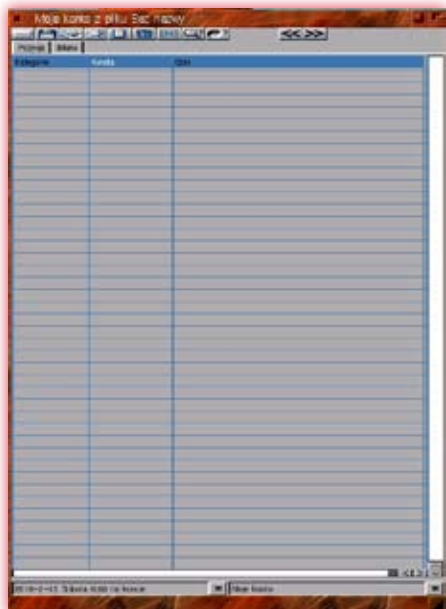
Wróćmy zatem do naszego budżetu. Przykład jakim się posłużę, pokaże jak wprowadzać dane, przychody, wydatki, zaprezentuje jak skorzystać z wykresów i statystyk, a dzięki podliczeniu bilansu (nawet na dłuższy okres), będziemy mogli zmonitorować nasze domowe operacje finansowe, tak aby łatwiej było nam zdecydować, w której dziedzinie wprowadzić cięcia, celem zaoszczędzenia kilku groszy. Powiedzmy, że mamy przeciętne, konto osobiste w banku, i chcemy troszkę zapanować nad migracją środków.

Po uruchomieniu programu (czy to z dyskietki, czy po ówczesnej instalacji na twardej), zakładany zatem nowy dokument, wybierając moduł „Budżet”.

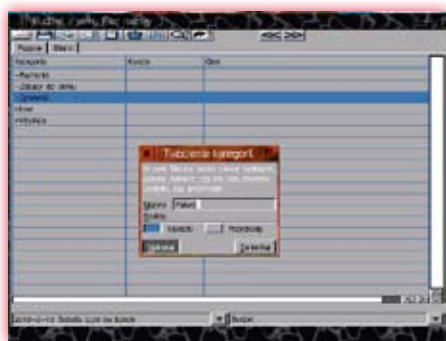


Ukazuje się naszym oczom okno programu, z tabelą podzieloną na trzy kolumny (załączka POZYCJE). Na samej górze mamy szereg klawiszy, przydatnych podczas korzystania z programu. Nie będę się tu skupiał na ich opisywaniu, gdyż ich funkcje są bardzo dobrze przedstawione przez dymki informacji pojawiające się po najechaniu na nie kursorem.

W KATEGORIA grupujemy nasze operacje finansowe. W zależności co wybierzemy tworząc grupę, czy będzie ona zaliczana do przychodów czy wydatków, program będzie dodawał lub odejmował wprowadzane, w kolumnie KWOTY liczby. Trzecia kolumna to OPIS. Tutaj wpisujemy komentarz, dotyczący wprowadzonej kwoty. Ułatwi nam to przypomnienie sobie, np. na co wydawaliśmy pieniądze.



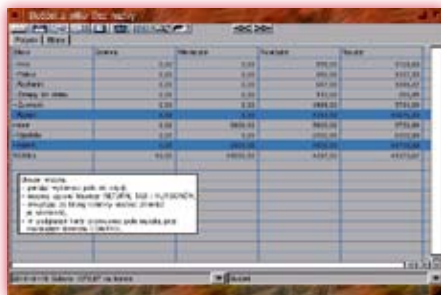
Klikając na pierwszą kolumnę (lub ikonkę z literą „N”), przejdziemy do okna tworzenia nowej, lub wyboru istniejącej już grupy przychodów, lub wydatków. Po wybraniu (lub utworzeniu) kategorii, pojawi się ona w pierwszej kolumnie. To czy będzie to przychód, czy wydatek, informować będzie także znaczek „+” lub „-”, przy nazwie. Warto dodać, że zestawianie listy kategorii, widoczne jest dla konkretnej daty. Jeśli wprowadzimy dane w jednym dniu, to podczas uruchomienia programu w dniu następnym, pierwsza kolumna będzie pusta. Żeby ją wypełnić należy znowu wybrać daną kategorię. Zapobiega to wprowadzaniu pustych danych, bo np. po co wypełniać całą kolumnę, jeśli w jednym dniu zrobiliśmy tylko zakupy, a w drugim tankowaliśmy tylko samochód.



Jeśli jesteśmy już na danych. Dzięki ikonom strzałek, możemy przesuwać kolejne dni, w przód lub w tył (np. żeby wprowadzić załagłe kwoty), lub posłużyć się kalendarzem, zaimplementowanym w programie.

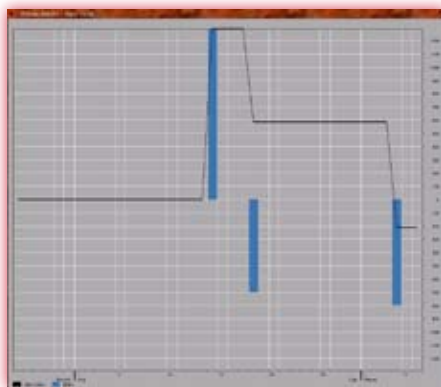
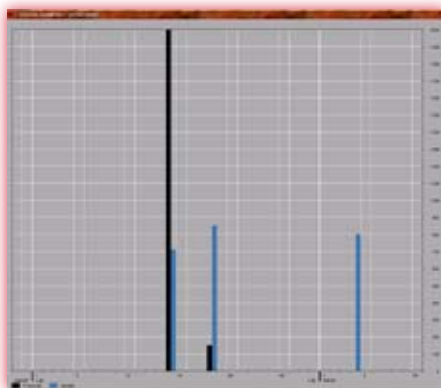
Podczas wprowadzania kwot, cały czas liczona jest suma wydatków i przychodów oraz ich różnica. Te dane można zobaczyć w zakładce BILANS. Dzięki niej będziemy mogli spojrzeć z szerszej perspektywy na nasze domowe operacje finansowe, gdyż bilans li-

czony jest dla konkretnego dnia, miesiąca, kwartału, oraz roku. Te dwie ostatnie pozycje mogą czytelnie przedstawić nam, jak radzimy sobie z prowadzeniem domowego budżetu, czy należy zacząć oszczędzać, czy też możemy pozwolić sobie na dodatkowy wydatek.



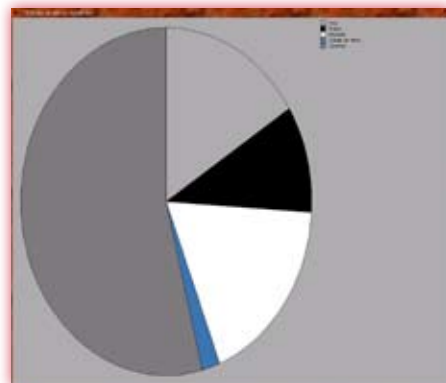
Kolejne opcje, opierające się także na danych z bilansu, w sposób graficzny (dla niektórych bardziej przemawiający niż cyferki), przedstawiają nam statystyki naszych poczynań z pieniędzmi.

Do wyboru mamy zatem: wykres przychodów i wydatków (słupkowy), struktura bilansu i stan naszej kasy (słupkowy z krzywą), oba dla całego miesiąca, oraz dwa wykresy kołowe. Pierwszy to wydatki, drugi to przychody. Przed ich rysowaniem, program pyta nas z jakiego okresu mają zawierać dane. Tak jak w bilansie, czy ma być to wykres zawierający dane z konkretnego dnia, miesiąca, kwartału, czy też z całego roku.



Aktualna kwota naszych pieniędzy (na konkretny dzień), cały czas wyświetlana jest w lewym dolnym rogu programu (obok ak-

tualnej daty), niezależnie od tego, czy jest ona dodatnia, czy też co gorsza, ujemna. Poza tym zapisany plik z naszym stanem środków, może zawierać także dane z dwóch pozostałym modułów. A przełączanie między nimi, podczas pracy z „Asystentem” odbywa się dzięki gadżetowi w prawym dolnym rogu aplikacji.



Na koniec dodam jeszcze, że możemy sobie wydrukować przeprowadzone operacje finansowe, w postaci wyciągu z bilansu, a cały plik obojętnie czy będzie zawierać dane dla kartoteki, terminarza, czy budżetu, można zabezpieczyć hasłem.

Umiejętność prawidłowego prowadzenia budżetu, czy to domowego, firmowego, czy też państwowego to zaleta bardzo pożądana w dzisiejszych czasach. A korzystanie z takich m. in. aplikacji może to ułatwić. Ale czy sam program wystarczy? Na to pytanie, nie da się jednoznacznie odpowiedzieć, stawiając np. naprzeciw sobie młode, dobrze prosperujące przedsiębiorstwo i dziurę w państwowym budżecie, która notabene, powstała za kadencji nie jednego parlamentu.

Dobra, dość tej polityki. Polecam program wszystkim tym, którzy chcą udowodnić swoim domownikom, że komputer (a zwłaszcza Amiga), nie jest zbędnym meblem pożerającym prąd, ale może być bardzo pomocnym narzędziem w zastosowaniach domowych.

Don Rafito

Gatunek: biurowy
Producent i wydawca: Twin Spark Soft
Rok: 1997
Ilość dysków: 1, HD
Platforma: każda Amiga, 1 MB Fastu,
 zalecany zegar podtrzymywany baterijnie

Ocena: 90%

Grimm



Amiga może pochwalić się wieloma aplikacjami, służącymi do edycji grafiki. Najbardziej znanymi są oczywiście takie programy, jak ImageFX, Art Effect, Personal Paint, czy też sztandarowy Deluxe Paint, choć te dwa ostatnie służą bardziej do tworzenia niż obróbki. Większość z nich nie jest już obecnie rozwijana, a część z nich zmieniła swój status, z komercyjnych na shareware, lub freeware.

Program który chcę zaprezentować, od początku był darmowy, a niektóre jego możliwości służące obróbce i edycji, mogą spokojnie konkurować z funkcjami z bardziej zaawansowanych programów do image processingu.

Po uruchomieniu, program otwiera swój własny ekran roboczy pracujący w 24 bitach. W zależności od sprzętu (AGA, lub karta graficzna), podczas pracy, obrazki wyświetlane są w kilku wariantach; 16 mln kolorów, HAM, Ord6 (256 kolorów), Ord2 (16 kolorów), GREY (256 odcieni szarości), oraz Multi. Ten ostatni podobny jest bardzo do trybu HAM, z tą różnicą, że jakość wyświetlanej grafiki jest nieco lepsza, ale za to paleta jest przyciemniona.

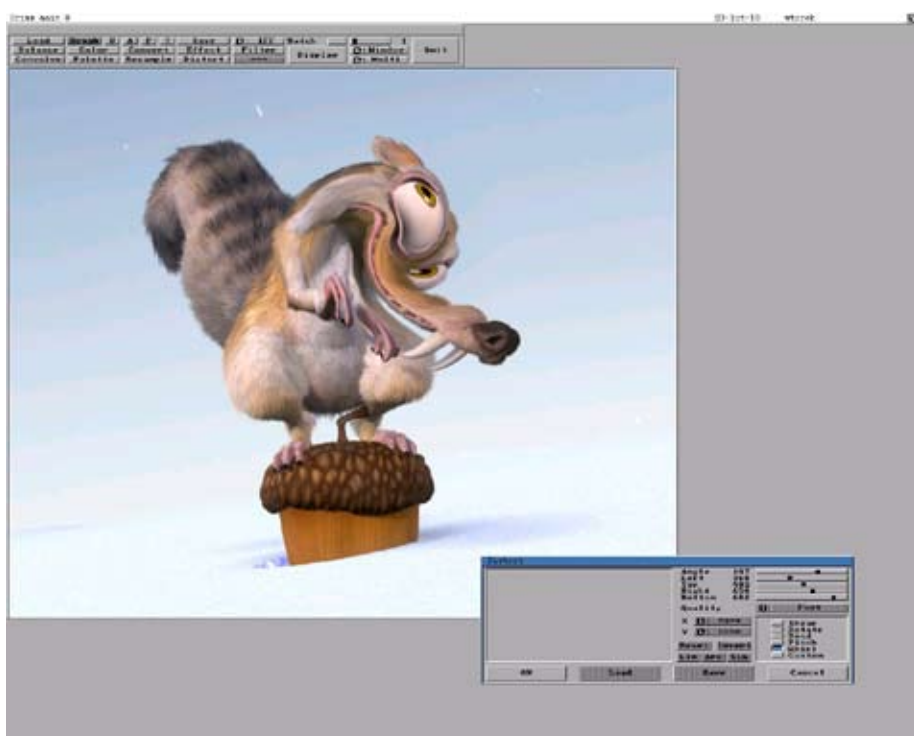
Ilość czytanych formatów jest imponująca:

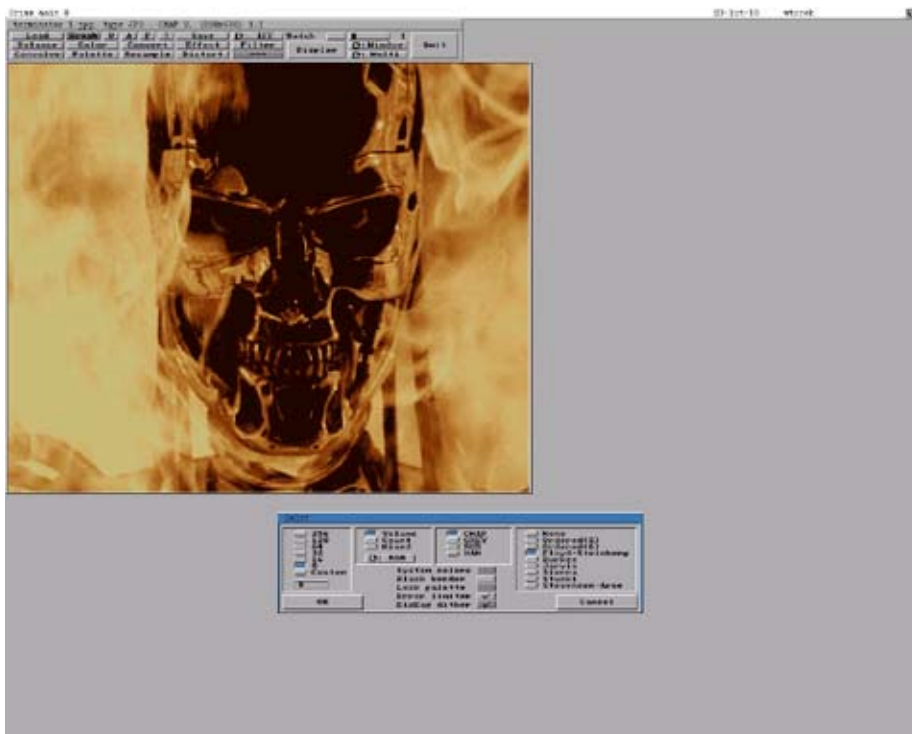
- BMP WIN-OS/2-RLE, 4/8/24,
- CIF w YUV 4:1:1,
- CCIR CCIR601 w YUV 4:2:2,
- DSF (własny format),
- FBM z pakietu Fuzzy BitMap,
- GIF 87 oraz 89
- IFF 1-8/24, HAM, PBM, SHAM, RGB8, RGBN, a także DEEP RGBM,
- JPG, JFIF
- PCD Base/16/4 oraz Base images,
- PCX 1/4/8/24 bity,
- P?M binarny oraz ASCII z biblioteką PBMPLUS.library,

- QRT (ray tracer) DKB lub POV,
- SUN 8/24/32,
- TGA 8/15/16/24/32,
- TIF 8/24
- PNG.

Zapis naszej pracy z obrazkiem, lub zdjęciem możliwy jest w jednym z kilku standardowych formatów:

- IFF,
- BMP,
- JPG,
- TGA,
- TIF,
- PPM,





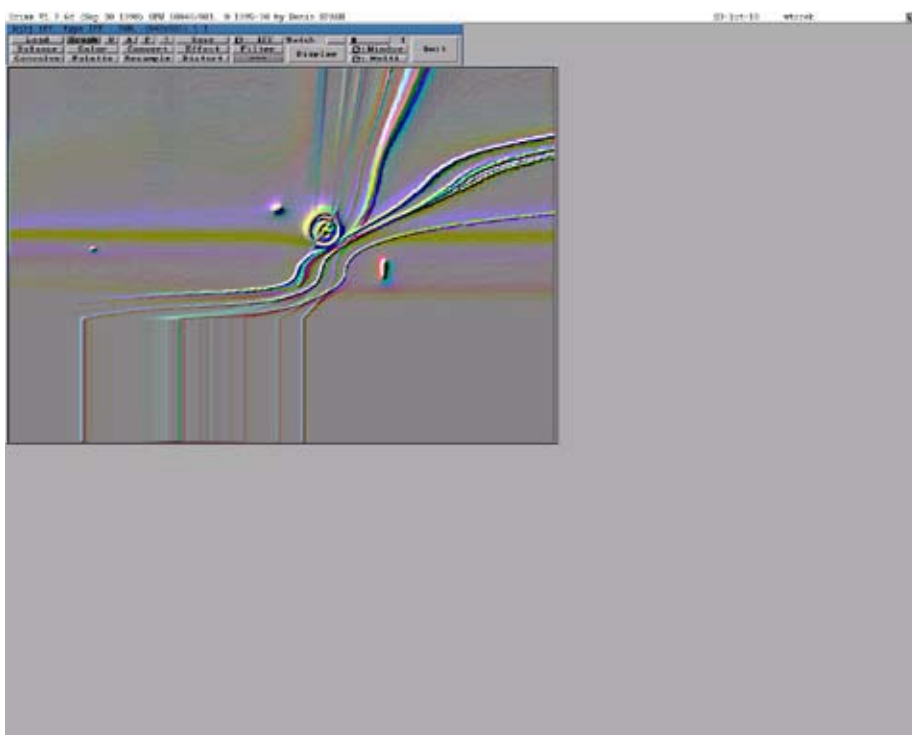
- DEEP,
- PNG.

Prócz samych grafik, „Grimm” obsługuje także kilka formatów animacji; Sequence, ANIM 5/7W/7L/8W/8L, FLC/FLI oraz nieskompresowane MPEG i AVI.

Wspomniany, dość ciekawy zestaw opcji pozwala nam na bardzo efektywną pracę z wczytanym zdjęciem, przynosząc ciekawe efekty, począwszy od zmiany rozmiaru i głębi koloru, poprzez nakładanie filtrów i poprawy jakości, na różnych ciekawych efektach i ewolucjach kończąc.

No ale po kolei. Pod klawiszem BALAN-

CE znajdziemy opcje służące regulacji składowych koloru (RGB lub YUV), nasycenia, kontrastu i jasności. COLOR służy nam do zmiany ilości kolorów w paletce obrazka. Z 24 bitów na HAM, lub CMAP. Opcja ta przydaje się, kiedy nie posiadając karty GFX, chcemy mieć fajny obrazek, lub fotkę z wakacji na blacie Workbencha. Zastosowanie kilku algorytmów rozpraszania (najbardziej znany to Floyd Steinberg), pozwoli nam z 24-bitowego zdjęcia zrobić dość dobrą tapetę, np. w 32 kolorach. W CONVERT kryją się narzędzia, pozwalające nam m. in. na obracanie i cięcie obrazka na kawałki. Takie efekty jak Oil, Mosaic, Posterize, Cube, Persective, czy Lens Flare, znane z wyżej wymienionych programów, znajdziemy



pod EFFECT. Pod przyciskiem FILTER kryje się zestaw filtrów; Minimum, Median, Maximum, Noise, czy Despckle. Opcja CONVOLVE to także filtry: Quick Edge, Kirsch, Pre-witt, Sobel, Laplacian, Sharpen, Smooth, czy Emboss, przy czym obrazek musi być w RGB (24 bity). Jeśli nie jest, program każe nam go przekonwertować do takiego formatu. A dołączona tabela, umożliwia edycję tychże filtrów, poprzez zmianę wartości w poszczególnych komórkach. W PALETTE wyświetlana jest aktualna paleta obrazka. Możemy w niej kopiować, przenosić, lub zmieniać poszczególne barwy i odcienie. Dzięki RESAMPLE możemy dowolnie zmieniać rozmiar wczytanej grafiki, także z użyciem algorytmów poprawiających jakość zmniejszanego, lub powiększanego obrazka. Jeśli mamy ochotę powyginać troszkę wczytaną grafikę, nałożyć wiry, lub wyrzucić fragment zdjęcia, to warto zajrzeć pod klawisz DISTORT. Prócz możliwości wyświetlania wczytanego zdjęcia w wyżej wymienionych trybach, po kliknięciu na DISPLAY możemy wyświetlić go na dowolnym ekranie i w dowolnej rozdzielczości dostępnej w naszej Amidze. Na koniec opcja BATCH. Dzięki suwakowi wczytany obrazek, można w dowolnym momencie powiększyć.

Do programu dołączono jeszcze trzy aplikacje: przeglądarkę obrazków „DSV” (obsługuje te same formaty co „Grimm”), oraz dwa odtwarzacze animacji; „A-IFF” (dla formatu IFF), oraz „A-FLIC” (format FLI/FLC).

Prezentowany program, prócz swojej wspomnianej darmowości, ma jeszcze kilka innych zalet. Ze względu na swoje gabaryty spokojnie można nazwać go pchełką. Archiwum waży niewiele ponad jeden megabajt. A jego niewygórowane wymagania dają możliwość odpalenia nawet na gołej A1200. I mimo że do efektywniejszej pracy z większymi obrazkami, czy zdjęciami, wystarczy już pamięć FAST i koprocessor (sam program jest w kilku wersjach, począwszy od MC68000, poprzez 020+FPU, na 060 kończąc), to i tak niewielka cena, za garść ciekawych opcji, dzięki którym nie tylko szary użytkownik, czy początkujący grafik amator, ale także bardziej zaawansowany znawca tematu znajdzie coś dla siebie.

Don Rafito

Gatunek: graficzny

Producent i wydawca: Denis SPACH

<http://dspach.free.fr/amiga/grimm/index.html>

Rok: 1998

Ilość dysków: archiwum około 1.2 MB, HD

Platforma: (praktycznie) 020, AGA, karta GFX

Ocena: 85%

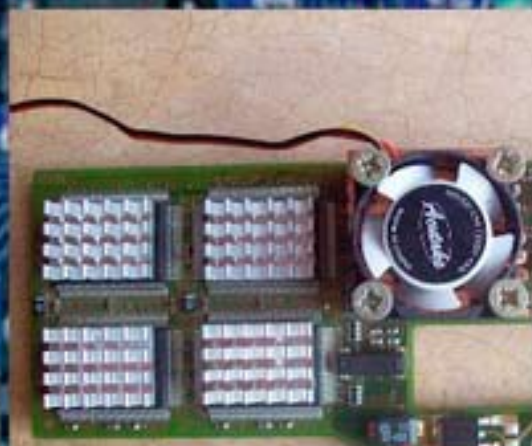


Polskie Pismo Amigowe

Numer 1/2009 (1)

cena 19 zł

NASZA RECENZJA



- ▶ Chłodzimy desktopa
- ▶ Stellarium
- ▶ AmigaOS 4 na A4000
- ▶ Efika w praktyce

Jakiś czas temu ukazał się czwarty i ostatni numer drukowanej gazety o Amidze, pod tytułem **Total Amiga**. Było to tłumaczenie angielskiego pisma pod tym samym tytułem, uzupełnione o teksty polskich autorów. Teraz, po przerwie, społeczność portalu **ppa.pl** ruszyła z inicjatywą stworzenia pisma dla Amigowców, składającego się w całości z tekstów polskich autorów.

Po głosowaniu na portalu wybór nazwy padł na nazwę **Polskie Pismo Amigowe**. Skrót pisma nawiązuje do nazwy **ppa.pl**, czyli Polskiego Portalu Amigowego. Pismo można nabyć wyłącznie za pośrednictwem tego portalu. Proces zamawiania przebiega bez zbędnych komplikacji. Potem już tylko przychodzi powiadomienie na skrzynkę e-mailową z numerem konta. Ponieważ pismo jest drukowane w niewielkiej liczbie egzemplarzy, może się zdarzyć, że na dodruk i wysyłkę trzeba będzie trochę poczekać. W moim przypadku jednak przesyłka dotarła bez żadnych opóźnień. Pismo przyszło starannie zapakowane w bąbelkową kopertę, która zapewnia dobre zabezpieczenie dla tego typu przesyłek. Z wielką niecierpliwością otwarłem kopertę i zabrałem się za lekturę. Cena gazety wynosi 19 zł. Zawiera ona koszt druku, pakowania i wysyłki. Jak zapewnia redakcja, pismo jest niedochodowe, a cena pokrywa jedynie koszty z nim związane. Ale dość już przynudzania – czas opisać zawartość.

Po wyjęciu gazety z koperty zaskoczyła mnie grubość pisma, a raczej jego „cienkość”. Przez długie lata amigowania przyzwyczaiałem się do grubszej prasy: albo ze względu na ilość stron, albo na grubość papieru. Polskie Pismo Amigowe składa się z 8 artykułów, komiksu na ostatniej stronie i działu wiadomości. Całość zajmuje 24 strony. Trzeba przyznać, że jakość techniczna pisma stoi na wysokim poziomie. Dobra jakość papieru i druk na profesjonalnych maszynach drukarskich ma na to znaczący wpływ i nie zmienia tego nawet fakt, że całość składana jest na OpenOffice 3. Jak w każdym piśmie tego typu, już na drugiej stronie znajdziemy apel o artykuły. Moi drodzy, nie ma się co czarować: życie gazet w tym segmencie zależy tak naprawdę głównie od ich czytelników, tu nie pojawi się żaden reklamodawca z artykułem przygotowanym przez jego dział marketingu. Pisma ukazują się, gdy czytelnicy zechcą pisać artykuły, a w przypadku ich braku los magazynów jest przesądzony. Dotyczy to także C&A Fan, które w tym momencie czytacie. Tak więc przyłączam się do apelu – **PISZCZIE ARTYKUŁY**. W chwili, gdy piszę te słowa, wiadome jest, że materiału zgromadzonego przez redakcję starczy jeszcze na jeden numer, ale dalsze losy pisma należą już do czytelników i redakcji.

Szata graficzna **PPA** (layout) jest na dobrym poziomie. Niezbyt bogata, ale za to przemyślana i czytelna. Poszczególne artykuły oznaczone są symbolem systemu, którego dotyczą. Dzięki temu od razu wiadomo, dla kogo jest przeznaczony dany artykuł. Oznaczenia te występują także w spisie treści. A czego dotyczą? Otóż trzy artykuły są dla użytkowników systemu 3.x, po dwa dla użytkowników MorphOS i AmigaOS 4.x, jeden artykuł jest polemiką.

Na początku dostajemy małą porcję newsów. Pozostaje pytanie, czy mając do dyspozycji Internet i np. portal **ppa.pl**, gdzie są one na bieżąco wystawiane, dział taki ma sens. No, ale redakcja zapewnia, że wybrano tylko te najważniejsze. Pierwszy artykuł jest z cyklu „zrób to sam” i opisuje boje autora z chłodzeniem rozbudowanej o kartę PPC i BVision Amigi 1200 w obudowie fabrycznej. Kolejny z kolei opisuje sposób rozpoznawania uszkodzonej pamięci na karcie turbo. Jest to wielce pouczająca metoda, podaje przepis na kompilację różnych potrzebnych programów ze środowiska PC na Amigę. Kolejny artykuł stanowi krótki przegląd trackerów dostępnych na Amigę klasyczną. I to wszystko odnośnie systemu 3.x.

Piotr Waligórski przedstawił na łamach **PPA** program działający pod MorphOSem pt. **Stellarium**. Jest to doskonały program astronomiczny i – co najważniejsze – darmowy. Omówiona została wersja 0.8.2; sposób jej instalacji, konfiguracji oraz podstawowe funkcje. Autor podpowiada jak korzystać z programu na Eфіce, gdzie występuje problem z ilością dostępnej pamięci.

W piśmie znajduje się jeden artykuł o charakterze polemicznym pt. „Komputer, to brzmi dumnie”. Traktuje on o historii rozwoju komputerów domowych i systemów na nie na przestrzeni ostatniego trzyletnia. No cóż, Panie Adamie, już w średniowieczu ludzie dostrzegli prawidłowość, że gorszy pieniądz wypierał z gospodarki lepszy. Tak samo jest i w naszym przypadku. Zazwyczaj tak jest, że technologia tańsza wypiera droższą i jej jakość ma tu bardzo drugorzędne znaczenie. Spójrzmy prawdzie w oczy. Główne potrzeby ok. 70% zwykłych ludzi oscylują wokół przeglądania netu i ściągania niezbyt wyszukanych treści, operowania kontem e-mailowym, do tego jakieś gadu-gadu i koniecznie odtwarzacz filmów + muzyki i gry. Do tego nie trzeba systemów w pełni konfigurowalnych i na wysokim poziomie informatycznym (zoptymalizowanych). Przeciętny Kowalski i tak nigdy do tych funkcji by nie dotarł i nie wykorzystał ich. On zwraca głównie uwagę na cenę. Dostrzegł to w porę jeden koncern i po prostu wykorzystał. Tych pozostałe 30% świadomych użytkow-

ników i tak nie ma siły przebicia liczonej w banknotach i dlatego jest jak jest. No, ale dosyć przynudzania, pora opisać kolejne artykuły zawarte w **PPA**.

Programiści PHP znajdą informację, jak pod systemem AmigaOS 4.x prawidłowo postawić serwer Apache oraz jak skonfigurować edytor VIM do pracy z kodem PHP. Artykuł kończy opis instalacji aplikacji do zarządzania bazą SQLite.

Kolejny artykuł wnikliwie opisuje zalety i wady Eфіki. Jest to doskonały artykuł dla osób zastanawiających się nad zakupem tego komputera. Opisane są problemy i wyzwania, przed jakimi stanie przyszły użytkownik. Autor obiecuje na zakończenie, że będzie kontynuował temat. Tak więc potencjalni użytkownicy nabydą solidną dawkę wiedzy o tym komputerze. Kto wie, może przyczyni się ona do popularyzacji tego sprzętu wśród Amigowców w Polsce. Choć na przeszkodzie może stanąć niewystarczająca dostępność sprzętu.

Ostatni artykuł traktuje o instalacji nowego systemu o numerze 4 na Amidze 4000. Autor stworzył typową instrukcję uzupełnioną o własne spostrzeżenia, jak krok po kroku przygotować sprzęt do instalacji, rozpocząć ją i jak zarządzać partycjami dysku twardego.

Na ostatniej stronie znajduje się komiks z postacią dobrze znaną wszystkim amigowcom. Komiks jest bardzo profesjonalnie przygotowywany przez będącego również w naszej redakcji BagoZonde. Komiksowi towarzyszy też bardzo ciekawy pomysł. Otóż czytelnicy sami, za pomocą dostępnej na portalu ankiety, mogą decydować o dalszych losach postaci.

Na zakończenie chciałbym podziękować wszystkim osobom mającym wkład w powstanie opisywanego pisma. Chłopaki, naprawdę dobra robota, tak trzymać! Pismo przeczytałem od deski do deski i nie było się do czego przyczepić. Ciekawe artykuły, napisane w dobry sposób i kolorowy layout naprawdę pozytywnie świadczy o gronie redakcyjnym. Wiele artykułów w tytułach lub w zakończeniach wskazuje, że to jest pierwsza część i w następnym numerze możemy spodziewać się kontynuacji. Do wielu artykułów znajdują się linki dla osób chcących powiększyć swoją wiedzę na temat w nich poruszony. Po przeczytaniu pisma pozostał pewien niedosyt. 24 strony to trochę mało i chciałoby się więcej i więcej, czego sobie i całej redakcji życzę. Co najmniej 100 numerów.

MrMat

FPS'y na Amigę

część 2

Witam w drugiej, a zarazem ostatniej części cyklu, poświęconego najlepszym grom z gatunku FPS dla Amig klasycznych. Ostatnio nasza przygoda ze strzelankami 3D stanęła na roku 1995, a ostatnią opisywaną grą był „Gloom” wraz z jego późniejszymi odsłonami. Kolejne lata przyniosły Przyjaciółce tytuły, które – dzięki swojemu lepszemu zaawansowaniu technologicznemu – udowodniały, że można zrobić program z rozbudowanym silnikiem graficznym, niezłym dźwiękiem czy świetnie napisanym scenariuszem. I choć prezentowane poniżej gry potrzebowały do komfortowej pracy już nieco bardziej rozbudowanych konfiguracji: kart turbo (myślę, że 030 z Fastem to absolutne minimum), twardych dysków, napędów CD, w większości kości AGA, a przy niektórych tytułach przydałaby się nawet karta graficzna, to programiści nadal wykorzystywali konstrukcję procesorów z linii mk68.



Komputer jest w stanie prześcignąć człowieka w wielu dziedzinach. Potrafi dużo szybciej liczyć, przetwarzać i gromadzić informacje. Potrafi także przewidywać sytuacje. Tylko że w tym całym przewidywaniu opiera się na danych i wzorach matematycznych. A są sytuacje, w których nie zawsze dwa plus dwa da nam cztery. Człowiek, wyposażony w intuicję, może wtedy poddać taki przypadek optymalnemu osądowi. Maszyna tego nie dokona. A kierując się wynikami danych, może doprowadzić do katastrofy, czego przykładem może być historia wprowadzająca nas do gry „Nemac IV”.

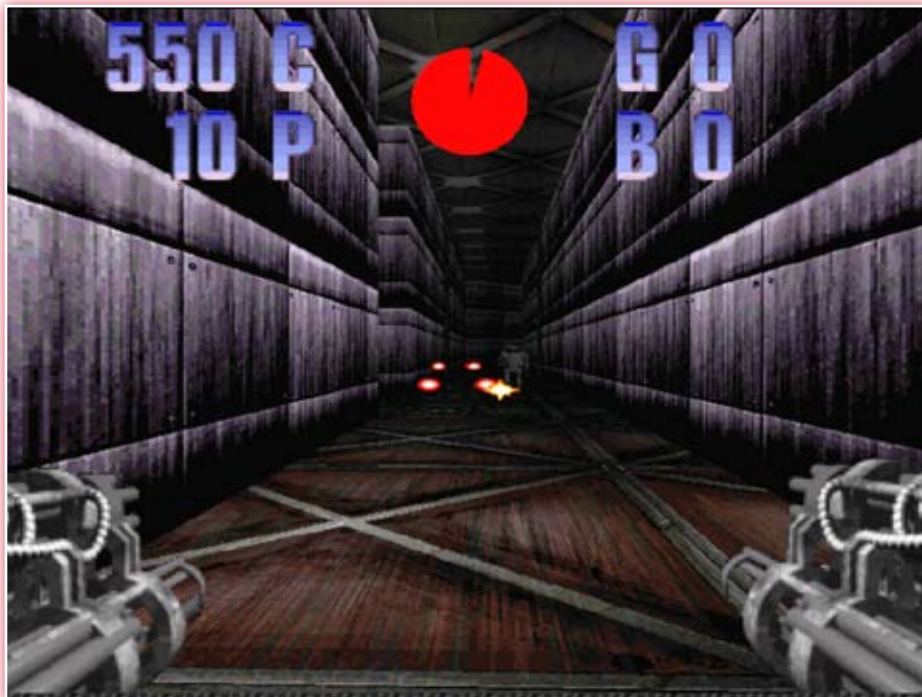
No, ale zacznijmy od początku. Mamy rok 2048. Trzy supermocarstwa postanawiają stworzyć globalny system obrony. Budując trzy połączone ze sobą bunkry, wyposażają je w neuronowe komputery, potrafiące w przeciągu sekundy przeanalizować obecną sytuację globalną i wydać konkretne dyrektywy.

Nazwano je odpowiednio: **Nemac I, II i III**. Ale niestety, skuteczność systemu obrony zarządzanego przez owe komputery okazała się niewystarczająca. Podczas wojny w latach 2057 – 2065 doszło do przeforsowania dwu z trzech bunkrów, zarządzanych przez **Nemac’a I i III**. **Nemac II** wykazał się w tym przypadku postawieniem prawidłowego osądu, co uchroniło bunkier przed zniszczeniami, ocalając życie wielu ludziom. Po wojnie sytuacja niewiele się poprawiła. Konflikt wisiał na włosku. Wyciągając wnioski, postanowiono stworzyć jeden, centralny system obrony, całkowicie zależny od najnowszego modelu superkomputera neuronowego. Owemu projektowi nadano nazwę **Nemac IV**. Bunkier wchodzący w skład przedsięwzięcia składał się z kompleksu pomieszczeń rozsianego kilkaset metrów pod powierzchnią Ziemi. Całkowicie zautomatyzowany system obrony superkomputera powodował, że przejęcie władzy nad **Nemac’iem IV** graniczyło z niemożliwością. Nawet przy zniszczeniu po-

szczególnych elementów siatki neuronowej, zadania natychmiast były przejmowane przez inne ocalałe fragmenty. Dodatkowo system nawigacji satelitarnej dawał **Nemac’owi** całkowitą kontrolę nad ewentualnymi posunięciami militarnymi. Nieograniczona kontrola nad bronią nuklearną, totalna niezależność oraz wymienione wcześniej cechy stwarzały z **Nemac’a IV** broń doskonałą. Ale czy skuteczną?



Aby się o tym przekonać, jego twórcy postanowili zasymulować sytuację ataku. Tylko nikt nie przewidział tego, że najpierw należało powiadomić o teście komputer, który po otrzymaniu danych i przetworzeniu ich wydał osąd: wojna! Człowiek, wyposażony właśnie w intuicję, dwa razy by się zastanowił, zanim cokolwiek by podjął. Cały ludzki personel, znajdujący się wewnątrz kompleksu, natychmiast został zdiagnozowany jako wróg. Nielicznym – zanim zginęli – udało się nieco osłupić **Nemac’a**, pozbawiając go m.in. podglądu sytuacji wewnątrz bunkra. Pozostał mu m.in. wspomniany zautomatyzowany system obrony, składający się z szeregu maszyn bojowych oraz możliwość odpalenia pocisków jądrowych, czego skutkiem mogłaby być totalna katastrofa. Co robić? Należy jak najszybciej dostać się do centrum superkomputera i go dezaktywować.



wać. Ale człowiek, jako forma życia, nie jest mile widziany w bunkrze i każdy, kto się tam dostanie szybko podzieli los całego personelu.

Postanowiono wprowadzić do bunkra coś, co by nie podpadło Nemaćowi, coś, co by nie było mu obce – maszynę. Podczas testów jednostki bojowej FIKM-7, wyposażono ją w zdalne sterowanie. Służyła także niszczeniu maszyn bojowych znajdujących się poza kontrolą. I to na ten egzemplarz padł wybór, gdyż wierzono, że nie zwróci ona uwagi na Nemać'a, który otaczał się właśnie maszynami.



Cel jest prosty: dojść zdalnie sterowanym FIKM-7 aż do samego reaktora, zasilającego zarówno Nemać'a, jak i cały bunkier i wysadzić go. Ale czy wykonanie rzeczywiście będzie takie proste, to się jeszcze okaże.

Zadaniem gracza jest właśnie zasiąść za sterami naszej zdalnie sterowanej zabawki, doprowadzić ją do reaktora i zapobiec globalnej katastrofie. Na drodze do celu spotykać będziemy niezbyt przychylnie nastawione jednostki bojowe, które po kontakcie „tworzą w twarz” szybko się orientują, że nie jesteśmy jednymi z nich i – wykonując swoje zadanie – przypuszczają na

nas zmasowany atak: wszelkiego rodzaju rody, roboty jeżdżące, latające... Krótko mówiąc, cały arsenał poruszającego się żelastwa.

W przeciwieństwie do wielu strzelanek 3D, gdzie musieliśmy wejść w posiadanie różnorodnej broni, tutaj mamy do dyspozycji cały możliwy asortyment od razu do wykorzystania: karabin, rakiety czy też granaty detonowane na odległość. Po drodze zbieramy amunicję oraz apteczki reperujące uszkodzenia w naszej zdalnej jednostce. Co prawda, apteczki trochę tu śmiesznie wyglądają (ciekaw jestem, jak wygląda taka naprawa; chyba tak, jak w naszej służbie zdrowia: jeśli się coś zepsuje lub połamie, zawsze można posklejać to plasterem). Poruszając się po kompleksie szukamy odpowiednich kodów dostępu w terminalach, którymi otwieramy poszczególne drzwi. Jeśli takowe nie wymagają kodu, to należy poszukać przełącznika. Poza tym przydatna może tu być mapa, wyświetlająca się nam bezpośrednio na ekranie. Możemy sobie ją powiększyć lub pomniejszyć. To, co w grze rzuca się w oczy w pierwszej kolejności, to ładne, renderowane intro, choć nieco rozpikselowane. Druga sprawa to jakość tekstur. Są one naprawdę dobrze wykonane, czytelne i wyraźne. Odpowiednio budują klimat w grze. I chodzi mi nie tylko o ściany, podłogi czy sufit. Wszelkiego rodzaju elementy są pokryte porządnie wykonanymi bitmapami. To też dotyczy naszych oponentów, którzy w większości są wektorowymi bryłami, a nie płaskimi, ruchomymi obrazkami. Nieźle są także niektóre efekty specjalne, np. wybuchy. Engine 3D już tak doskonały nie jest, mało tego: nie jest całkowicie 3D. Mimo zastosowania różnych pochyłych krzywizn i innych ciekawych atrakcji, cały czas poruszamy się po jednej płaszczyźnie. Wynagrodzić

to może duża możliwość konfiguracji ekranu gry, co ucieszy użytkowników nierozbudowanych Amig, i to niekoniecznie wyposażonych w AGAtkę, ponieważ gra wystartuje także na ESC'ie (wystarczy tylko procesor 020, 2MB pamięci graficznej i około 7MB wolnego miejsca na twardym dysku). Co prawda 2x2 piksel i zmniejszona wielkość okna nie jest imponująca, ale gra ruszy. Ilość klatek animacji wszystkiego, co można zobaczyć na ekranie także jest dobra i stwarza wrażenie realizmu.

Strona dźwiękowa jest zdecydowanie piętą achillesową programu. Poza niezłym modulem podczas intra, nic już nie zaszokuje pozytywnie naszych uszu. Programiści w tej kwestii poszli na całkowitą łatwiznę, adaptując do „Nemać'a IV” nie swoje, na dodatek marne, efekty dźwiękowe (np. część sampli pochodzi z gry „Steel Empire”, a część z bibliotek public domain). Gdyby nie wspomniane wyżej intro, byłaby „pała” z wykrzyknikiem.



Z nowinek technicznych to fakt możliwości skorzystania z karty graficznej oraz, co jest już całkowitym novum, z hełmu wirtualnego. Jeśli dysponujemy akurat takowym sprzętem, to wrażenia mogą być niesamowite. Szkoda, że w grze zabrakło opcji multiplayer. No, ale wszystkiego mieć nie można. Reasumując, grę można sklasyfikować tak gdzieś w środku amigowych FPSów. I choć w „Nemać'a” gra się całkiem przyzwoicie, pomijając fakt fatalnego udźwiękowienia (zawsze można zapuścić sobie swoją ulubioną muzykę w tle), to nie jest ona przeznaczona tylko dla fanów s-f (z racji swojego charakteru), ale dla wszystkich, którzy szukają w FPS'ie dobrej fabuły, niezłej grafiki oraz ciekawej i dynamicznej akcji.

Gatunek – FPS s-f
Producent i wydawca – Zentek
Rok – 1996
Ilość dysków – 8, HD
Platforma – 020, ESC, AGA
OCENA:
 Grafika – 85
 Dźwięk – 45
 Grywalność – 80

Ogółem: 70

Alien Breed 3D 2

– The Killing Ground



Po ostatnim survivalu, którego historię opisuje pierwsza część tej gry, nasz bohater nieświadom niczego, w błogim śnie hibernacyjnym, wracając do domu, zostaje ściąnięty przez Obcych. Gdy krążownik zadokował, wybudzony Reynolds doznaje szoku. Widok zwłok żołnierza oraz rozchlapanej, zielonej mazi daje mu do zrozumienia, że jego koszmarny jeszcze się nie skończył. Że przyjdzie mu znowu walczyć o przeżycie. Jedyne, co mu zostanie w chwili klęski, to śmierć. Nie ma więc wiele do stracenia i bez walki się nie podda. Chwyta za leżącą obok, słabo naładowaną strzelbę i rusza naprzód.

Rok po ukazaniu się pierwszej części „Alien Breed 3D”, Team 17 postanawia ucieleśnić kontynuację tej wspaniałej strzelanki 3D. Tym razem akcja toczy się na planecie – statku, zamieszkałej przez naszych oprawców, co może sugerować podtytuł – „Zabójcza Planeta”. Do przemierzenia znowu mamy 16 poziomów, do granic możliwości nafaszerowanych najróżniejszymi rodzajami robotów, wielgaśnych owadów, potworów i oczywiście znanych już nam czerwonych diabełków. Część z nich zamieszkują strażnicy, grube ryby. Rozwalenie ich umożliwi nam nie tylko ocalenie własnej skóry, ale także kontynuację morderczej wędrówki. Do dyspozycji mamy podobny względem pierwszej części zestaw

bronii: od strzelby, poprzez blaster, plazmę, granaty, miny, na wyrzutni rakiet kończąc. Zadbano także, aby zestaw asortymentu nie był bogaty: mało amunicji, mało apteczek, w ogóle wszystkiego mało, co może nam pomóc przeżyć i wyjść cało z opresji. Chociaż nie do końca. Jedną z ciekawszych rzeczy, którą wprowadzono jest plecak antygravitacyjny, pozwalający

nam na latanie. Na niektórych poziomach jest on wręcz niezbędny, aby móc ukończyć etap, gdyż wyjście, którego szukamy, znajduje się na półce, do której nie dojdziemy piechotą.

Przemierzanie całego kompleksu korytarzy w poszukiwaniu kluczy otwierających poszczególne drzwi łatwe nie jest. Architektura jest tak rozbudowana, że gdyby nie samorysująca się mapa, wyświetlana na ekranie, nasze błądzenie po poziomie mogłoby trwać bez końca. Przy okazji moglibyśmy trafić na bandę robotów. A tu co? Dwa naboje w strzelbie. Trzeba zwiewać, no ale czy to ma sens? Nasi oprawcy, wyposażeni w świetną sztuczną inteligencję i tak nas znajdują. Jak nie będą mogli do nas dotrzeć jedną drogą, to znajdą inną. Mało tego, przyprowadzą ze sobą kumpli. Jednym słowem, przerąbane.

Poza tym w grze wprowadzono kilka ciekawych innowacji, starając się zwiększyć jeszcze bardziej realizm, jednocześnie utrudniając nam do maksimum próbę przeżycia podczas tej całej rzeźni. Np. aby powystrzelać, co się da, nie zawsze wystarczy jedno małe pif-paf. Czasami trzeba zużyć nawet kilka naboju na pojedynczego robota. Trzeba zatem celnie trafiać i maksymalnie oszczędzać amunicję. Prócz tego wprowadzono efekt odrzutu podczas trafienia. A co tam, że przez pół godziny szukaliśmy po



labiryncie paliwa do naszego plecaka, a startując w górę w ostatniej chwili uszliśmy z życiem przed śledzącym nas od pewnego czasu robotem, którego kroki rozlegały się po całym labiryncie. Dostajemy przysłowiową kulkę w łeb, a odrzut zrzuca nas wprost pod stopy naszego oprawcy.

To, co w pierwszej części gry kulało, tutaj zostało naprawione, poprawione i usprawnione. Pełny trójwymiarowy silnik graficzny, wszelkie możliwe płaszczyzny i krzywizny, wszystko pokryte bardzo dobrymi, odpowiednio stonowanymi kolorystycznie teksturami. Refleksy światła (np. korytarz, w którym panuje półmrok, nagle zostaje rozświetlony lampami zbliżającej się maszyny), przezroczystość wody, szczegółowość elementów, widok broni, efekty pirotechniczne. Nawet nasi wrogowie są trójwymiarowymi bryłami. Krótko mówiąc, grafika w grze to absolutna rewelacja. A dzięki możliwości zmiany ustawień wyświetlanego ekranu, od rozmiaru okna, po szczegółowość grafiki, odpalić ją mogą także posiadacze słabszych Amig.



Z animacją jest już troszeczkę gorzej. Mimo dobrej liczby klatek zdarza się, że gra



zwalnia i szarpie przy dużej liczbie obiektów, nawet na rozbudowanych konfiguracjach.

Co do strony dźwiękowej, to pozostano tu przy standardzie wprowadzonym w pierwszej części, nieznacznie go ulepszając. I tak dla przypomnienia: możliwość ustawienia czterech i ośmiokanałowego dźwięku, zarówno mono, jak i stereo. Wszelkie efekty, strzały, kroki, wybuchy, dźwięk otwierających się drzwi, są dobrze wykonane i realistycznie brzmiące. Dopelnieniem tego może tu być muzyka. Choć nie usłyszymy tu konkretnych modułów, przypominających piosenkę, to zmieniające się w zależności od sytuacji, w jakiej się znajdziemy utwory, przypominają te z najlepszych kinowych horrorów.

„The Killing Ground” został wydany w dwóch wersjach. Pierwsza, przeznaczona dla Amig wyposażonych tylko w 2 MB pamięci graficznej, ma zubożoną grafikę (tylko tryb 2x2 piksel) oraz tryb ośmiokanałowego dźwięku. Natomiast druga przeznaczona była dla komputerów wyposażonych dodatkowo w pamięć Fast (min. 4 MB). Poza pełnymi walorami graficznymi – dźwiękowymi, zawiera ona także edytor, w którym możemy sobie zaprojektować poziom.

Wspaniała fabuła, świetna grafika, trzymający w napięciu klimat grozy, podsyćany przez różnymi dźwiękami i bardzo duży realizm tworzą grę, jakiej jeszcze nie było na Amidze do tej pory. I pomimo tego, że jest ona naprawdę trudna, sprawiająca wrażenie, że dedykowano ją tylko zaawansowanym graczom faktu, że nie ma na świecie Amigi (obojętnie w jaką Motorola by nie była wyposażona), na której „Alien Breed 3D 2” chodziłby niezawodnie płynnie, to jest ona pozycją obowiązkową dla fanów FPS-ów trzymających w napięciu. Pozostałym zostaje tylko popatrzenie z zazdrością na ekrany ich kolegów, którzy z przyspieszonym biciem serca próbują przeżyć na tej „Zabójczej Planecie”.



Gatunek – FPS s-f
 Producent – Team 17
 Wydawca – Ocean
 Rok – 1996
 Ilość dysków – 4
 Platforma – 020, AGA
 OCENA:
 Grafika – 95
 Dźwięk – 95
 Grywalność – 80

Ogółem: 90

Trapped, Trapped 2



Jak do tej pory rozprawialiśmy się nad grami bazującymi na wątkach s-f, strategii czy przygodówek. Omówione poniżej dwa tytuły to jedne z pierwszych, które były trójwymiarowymi wersjami role-playing. Tak więc, jak przystało na RPG'a, prócz strzelania w 3D, mamy tu rozbudowaną fabułę, klimat fantasy, magię, czary, możliwość rozwijania się naszej postaci i wiele innych cech tego gatunku, w którym nie liczy się przycisk FIRE i parcie do przodu.

Ale do rzeczy. Pokonany i uwięziony w katakumbach bóg Tarnak, jak Feniks odradza się i znowu zaczyna zagrażać mieszkańcom Kaldrionu. Nie pozostaje nic innego, jak rozprawienie się z nim. Aby tego dokonać należy użyć koła Talmara. W przeciwieństwie do poprzedniej kreatury, jest to dobry bóg magii. Jest tylko jeden problem: owe koło trzeba najpierw pozbić do kupy. Jego dwanaście części rozrzucono po kompleksie podziemi, zamieszkałych przez najgorsze monstra, które niezbyt chętnie oddadzą poszczególne fragmenty. Ostatni poziom, trzynasty, to zmierzenie się z samym Tarnakiem. Wiele śmiazków próbowało tego dokonać i tyleż samo poległo. Cała nadzieja w Tobie, miły gracz.

Przed rozpoczęciem gry wybieramy sobie zatem postać. Każda charakteryzuje się innymi zdolnościami, siłą fizyczną, zdolnością posługiwania się bronią, magią oraz wytrzymałością.

Doświadczenie, sprawność fizyczna i magia to nie wszystko. Nasza postać może posługiwać się różnymi rodzajami broni białej, która też nie będzie uniwersalna dla wszystkich, a skuteczne posługiwanie się nią będzie uzależnione od granej przez nas postaci. I tak do dyspozycji mamy: sztylet („dagger” - słaba, ale i lekka broń, nie zadaje strasznych ran, najlepiej sprawdza się w dłoniach myśliwego), długi miecz („long sword” - nieco ciężkawa, najlepiej władać nią będą rycerz i wojownik), morgernstern („mornig star” - szkoda że nie łańcuchowy, dobry dla postaci o dużej sile i umiejętnościach, np. dla rycerza), młot („hammer” - przeznaczony dla kulturystów, barbarzyńcy i krasnoluda), trójjąb („pike” - broń o dalekim zasięgu i dużej szkodliwości, najlepiej posługiwać się nią będzie myśliwy), topór („war axe” - skuteczna, ale i ciężka broń, mogąca być dobrze użyta przez krasnoluda). Zabrakło tu niestety czegoś, co działałoby na odległość, np. kuszy czy łuku. Co do wspomnianej magii, każdy czar potrzebuje jakiegoś paliwa. Tutaj tym paliwem jest zebranie odpowiedniej ilości i jakości fogów (magicznych kamieni). Znalezione papirusy opisują, jakich kamieni i w jakim zestawieniu użyć, aby wywołać pożądane zaklęcie. Wartą wspomnienia ciekawostką jest to, że po zebraniu fogów, po jakimś czasie znowu się pojawiają w tym samym miejscu. A zebranie jak największej ilości ograniczone jest tylko pojemnością naszych kieszeni.



Znajdowanie kluczy, przełączanie przycisków w ścianie czy postawienie jakiegoś przedmiotu w punktowym miejscu to główne sposoby na przechodzenie do kolejnych etapów gry. Pomocna jest tu także mapa, której wywołanie zazwyczaj otwiera nam osobny ekran (wyjątkiem może być gadżet, który, choć na krótko, ukazuje nam mapę na ekranie rozgrywki).

Nasi przeciwnicy to niezbyt sympatyczni kolesie, jakieś dziwne monstra, przerośnięte pająki, zombie itd., obdarowani w nienajgorsze algorytmy. Najpierw napierają na nas z maksymalną siłą, a po tym, jak zrobi im się ciepło pod stopami, czmychają w nadziei, że sobie odpuszczimy i darujemy im życie. A wtedy wystarczy tylko moment, żeby znowu przyspuściły atak.

Mimo, że „Trapped” nie jest wykonany w pełnym 3D (choć znajdziemy tu takie atrakcje jak schody), to dzięki pewnym rozwiązaniom nie jest to aż tak zauważalne. Mowa tu przede wszystkim o różnorodnych obiektach, chyba po raz pierwszy wprowadzonych w FPS’ach. I tak spotkać możemy tu stoły, krzesła, liczne pułapki wysuwające się wprost z podłogi, wyrzutnie ognistych kul czy kręcących się kółczatek. Gra zyskuje dzięki temu swoisty charakter i klimat. Tekstury są dość czytelne, choć czasami zauważyć można lekkie rozpikselowanie. Przeróżne efekty: refleksy światła, cieniowanie, rozmazanie także są na dobrym poziomie i nieźle się prezentują.

Natomiast niezbyt ciekawie rozwiązano tutaj sposób wykonania wrogów. Zamiast trójwymiarowych brył, walczymy tutaj z ruchomymi bitmapami, mającymi na dodatek niezadowalającą ilość klatek animacji. Bardzo



to utrudnia walkę z potworami, które zamiast poruszać się płynnie skaczą po ekranie, jak ludzik z kiepskiej gierki LCD. Zabrakło tu także widoku broni.

Zadawała natomiast duża możliwość konfiguracji wyświetlania. Prócz samodzielnego grzebania w ustawieniach (szczegółowość grafiki, włączanie i wyłączanie niektórych efektów itd.), możemy dobrać odpowiedni model lub wymusić na programie autodetekcję sprzętu. Zaznaczyć także trzeba, że gra jest w wersji tylko dla Amig wyposażonych w procesor 020 i 2 MB pamięci graficznej.

Nieźły moduł podczas wprowadzenia to niestety ciut za mało na dobre noty dla strony dźwiękowej. W dalszej części i podczas samego

grania towarzyszyć nam będą już tylko efekty dźwiękowe. Żadnych modułów i melodyjek, które w jakiś sposób podniosłyby klimat w grze. Kroki, odgłosy naszych wrogów, szcęk używanej broni, jakieś śpiewy ptaków w tle są raczej średniej jakości. Większość sampli brzmi dość syntetycznie.

Tyle, jeśli chodzi o pierwszą część gry. Programiści, nie owijając w bawełnę (co i ja uczynię) rok później postanawiają wydać jej kontynuację.

Zły Tarnak, mimo sukcesu śmiałka, który składając koło Talmara unieruchamia go na tysiąc lat, postanawia pokrzyżować plany mieszkańców wspomnianej wcześniej Kadrion. Czmychając ze świątyni (w której notabene był więziony) zabija kapłanów – strażników i znajdując schronienie w Pałacu, zsyła na biednych ludzi najgorsze rodzaje potworów. Ci, z przerażeniem w oczach, rychło postanawiają ewakuować się z obleganej poczwarami miejscowości, często w wielkim pośpiechu, porzucając cały dorobek życia. Ci, którzy postanowili jednak się bronić, szybko poginęli w nierównej walce. Tym razem trzeba się z gościem rozprawić ostatecznie. I to zadanie powierzone zostaje nam. Nie możemy zawieść biednych mieszkańców Kadrion. Jesteśmy ich jedyną deską ratunku.

Jak się można domyśleć, cała akcja prowadzona będzie w czeluściach owego Pałacu. Do dyspozycji mamy tutaj tylko jedną postać, a nie (jak w pierwszej części) kilka, która podczas przechodzenia przez kolejne siedemnaście etapów, rozwija nam się pod względem umiejętności, doświadczenia i wytrzymałości. Zbieranie magicznych artefaktów, rozwiązywanie zagadek, przełączanie przycisków czy



szukanie kluczy to jedno z głównych sposobów na dotarcie do pomysłodawcy całego zamieszania. No, ale żeby nie było tak łatwo, Tarnak porozstawiał po całym kompleksie najróżniejsze poczwary i potwory, poczynawszy od różnego rodzaju pajaków, skorpionów, os, na czelkowsko kształtne zombie kończąc. Do dyspozycji mamy, podobny jak w „jedynce”, zestaw broni białej, różniący się od siebie wyglądem, zasięgiem czy szybkością. Poza tym, jako nowość wprowadzono możliwość stosowania zaklęć ofensywnych. Dzięki temu magia daje nam nie tylko moc uzdrawiania, wzmacniania siły czy dodawania większej ilości gadżetów do naszego ekwipunku (mapa, kompas), ale także pozwala eliminować naszych wrogów oraz udoskonalać posiadaną broń. Poza tym, zbieranie i popijanie magicznych płynów w kolorowych flakonikach także okazuje się przydatne.

W grafice także zmiany. Bardziej rozbudowany engine 3D, zawierający już nie tylko schody, ale także półki, różne krzywizny i płaszczyzny, przestrzenne komnaty z wysokim sklepieniem, a także wąskie i ciemne korytarze. Wszystko pokryte ładniej wykonanymi teksturami. Do tego znane już nam krzesła i stoły, różne pułapki, wystające kolce, ściany ognia czy nawet widok broni. Refleksy światła i efekt flary, lustra i najróżniejsze efekty zabaw ze szkłem dają wrażenie kolorowego i krystalicznego obrazu w grze. Dodatkowo, renderowane animacje, pojawiające się podczas rozgrywki, nie tylko odsłaniają fabułę, ale także dopełniają dobrą stroną graficzną gry.

Zmieniło się także wykonanie wszelkich kreatur. Choć zrobione są one z wektorowych brył, to jakość tekstur, jakimi je pokryto, pozostawia wiele do życzenia. Są mało czytelne

(ciężko się np. dopatrzeć, gdzie zombie ma oczy czy nos) oraz nieco „krowiaste”. Poza tym, przy niektórych kątach patrzenia, ulegają częściowemu „wchłonięciu” przez scenę. I tak np. zamiast całego pajaka, widać tylko jego głowę i wystające przednie odnóża. Pozostałość jego ciała tkwi w ścianie. Płynność ruchów nieco zyskała, ale ilość klatek animacji nadal jest niewystarczająca.

Prędkość programu zależy przede wszystkim od sprzętu, jakim dysponujemy, choć tu gracze, dzięki dużym możliwościom konfiguracji, powinni być zadowoleni. Możliwości manipulacji są naprawdę spore: od zmiany rozdzielczości ekranu i wielkości okna, poprzez zabawy z jakością tekstur, wyłączanie i włączanie efektów (m.in. wspomniane refleksy światła czy też zabawy ze szkłem), na wyłączeniu renderowanych wstawek kończąc. Podczas tychże zabiegów na preferencjach ekranu wyświetlana jest informacja, na jakiej Amidze (rodzaj procesora oraz ilość pamięci) gra ruszy z wybranymi parametrami.

W kwestii dźwięku niestety zmian jest niewiele. Przynajmniej tych na lepsze. Jest jeden całkiem ciekawie brzmiący utworek podczas intra, a cała reszta podobna jak w pierwszej części. Sample i efekty dźwiękowe może nie brzmią już tak sztucznie, jak w „Trapped”, ale nadal są słabej jakości.

Biorąc pod lupę grywalność, w obu częściach momentami ona kuleje. A to dzięki fatalnemu rozwiązaniu sterowania w grze. Wiadomo, RPG powinien być rozbudowany, ale



podstawowe ruchy postaci czy kilka głównych form ataków i czarów, powinny być realizowane w kilku kliknięciach. A tak, zamiast tego, aby poruszać się dość sprawnie, trzeba wiele czasu poświęcić na rozpracowanie klawiszologii. Poza tym, jeśli używamy myszki, nasz bohater czasami zaczyna kroczyć własnymi ścieżkami i aby go przywrócić do porządku, trzeba poklikać w kursory, przy okazji obrywając w łeb od jakiegoś potworka.

No, ale dosyć bluzgów. Prezentowane tytuły może troszkę przez przypadek dostały się do naszego cyklu, gdyż ciężko nazwać je rasowymi FPS'ami – są raczej miksem FPP i RPG. Niemniej jednak są warte polecenia, szczególnie dla fanów gatunku. Jeśli Twoje ambicje są nieco większe niż tylko łażenie po trójwymiarowym świecie i strzelanie do wszystkiego, co się rusza, jeżeli lubisz łamigłówek, zagadki i klimat fantasy, to te gry są dla Ciebie. Niezła grafika, ciekawa fabuła oraz systematyczna nauka zasad panujących w „Trapped” i „Trapped 2” mogą zafundować godziną rozrywkę na wiele, wiele godzin.

Już na sam koniec, jako ciekawostkę można dodać, że programiści pracowali na trzecią część gry, ale nie została ona nigdy wydana. „Trapped 3” opary na silniku z dwójki odbiegał już od RPG. Klimat fantasy zamieniono na s-f, a miecze i magiczne zaklęcia zamieniono na broń palną, czyniąc grę bliższą gatunkowi FPS'ów.



Gatunek – FPS (a raczej FPP) RPG
Producent – Oxyrion
Wydawca – New Generation – Worldwide
Rok – 1996 (Trapped), 1997 (Trapped 2)
Ilość dysków – 7, HD (Trapped), 1 CD (Trapped 2)
Platforma – 020, OCS, ECS, AGA, karta GFX
OCENA:
 Grafika – 90
 Dźwięk – 60
 Grywalność – 70

Ogółem: 75

Brain Killer



Nigdy nie dokończone lub też nie wydane projekty często mają to do siebie, że nie są doceniane. Wiadomo, że dla wydawcy liczy się przede wszystkim zysk, a cała reszta niewiele go obchodzi. Zdarza się tak, że mimo to gry ukazują się, choć ze statusem freeware. Gorzej jest, gdy lądują w koszu. Takie właśnie perypetie spotkały się naprawdę dobrane, lecz wydawcy to nie wystarczyło.

Któregoś pięknego dnia polska grupa Virtual Madness postanawia zrobić FPS'a, który by nieco odbiegał od znanych standardów, przyjętych w strzelankach 3D. Tak rodzi się pomysł stworzenia programu pod nazwą „Brain Dead”. Podpisanie umowy z polskim wydawnictwem Alderan przyspiesza prace. Niestety, po ukończeniu gry, wydawca nabiera wody w usta, a zapal pro-

gramistów z grupy zostaje ostudzony zimnym prysznicem. Demo gry, zamieszczone w Aminecie zbiera bardzo przychylne opinie. Do tego stopnia, że FPS'em interesuje się niemiecka firma Titan. Niestety, umowa zawarta z poprzednim wydawcą sprawia, że „Brain Dead” jest nie do ruszenia. Autorzy, bazując na zdobytym doświadczeniu, tworzą „Brain Killer'a”. Umowa z Titan, tak jak poprzednio, napęla pomyślnymi wiatrami żagle programistów z Virtual Madness. Demo, pущzone w obieg, zbiera dobre recenzje. Amatorzy strzelanek 3D zacierają ręce: zapowiada się całkiem dobra pozycja. I to tyle dobrego, co można powiedzieć w tej kwestii. Nowy wydawca stwierdził, że ukończony program nie zadowala go. Że brakuje tego czy tamtego (m. in. czepiano się braku pełnego silnika 3D, pomimo tego, że cała reszta była obiecująca). Nakazuje przepisanie kodu od początku. Chłopakom z grupy nie za bardzo się to uśmiecha. A narastający brak czasu (część studiowała, inni powołani zostali do odbycia zasadniczej służby wojskowej) spowodował, że ostatecznie prace poprawcze nad grą zostały wstrzymane. Projekt ostatecznie trafił do kosza.

A szkoda. Wspomniane wyżej demo (na którym oprę swój dalszy opis) zapowiadało świetnego FPS'a, który po ukazaniu się wzbogaciłby kolekcję Amigowców o kolejną dobrze wykonaną i ciekawą grę.



Ni stąd, ni zowąd, nagle znajdujemy się w jakimś korytarzu. Do naszych uszu dobiega oddźwięk kroków, jakieś krzyki. Zza roku wyłania się koleś. Jego mina mówi sama za siebie. Wyciąga broń i zaczyna do nas strzelać. Nie trafia. Świszczące kule odbijają się od ścian, pokrytych jakąś blachą. Unikając kolejnych trafień podbiegamy do typa... i nagle się okazuje, że nie mamy mu się czym odwzajemnić. No tak, ale całkiem bezbronni nie jesteśmy. Porządny zamach i drań liże nasze podeszwy. Jeszcze dwa kopniaki, prawy sierpowy i nasz oprawca leży już nieruchomo na podłodze. Oglądamy się dookoła. Na razie spokój. Ruszamy dalej. Kurde, następny. Kolejny cios i nagle odkrywamy, że jesteśmy krzyżówką Stallone'a i Van Damme'a. Mimo wszystko przydałaby się broń. Dobra, jest. Uwaga! Miny. Mała wymiana ognia. Szukamy wyjścia. OK, następny poziom.

Broń, jaką posługuje się gracz w „Brain Killerze” to właśnie jedna z innowacji, wprowadzona do amigowych FPS'ów. Prócz całego strzelającego i wybuchającego arsenału, zadowalającego nawet wybredniejszych amatorów gatunku, nasz bohater może posługiwać się także własnymi butami i pięściami. Poza tym cel w grze, mimo pewnych ciekawszych rozwiązań, jest dość schematyczny. Powybijać wszystko, co się rusza, przeżyć i znaleźć wyjście. Zauważalny w grze klimat s-f jest tutaj nieco mroczny, co dobrze wpływa na odbiór.

To, co przede wszystkim zniechęciło Tita do wydania gry, to niepełny silnik 3D. Owszem, grając w „Brain Killer'a” poruszamy się tylko po jednej płaszczyźnie, brak schodów, półpięter, pólek itd., niemniej jednak, pozostałe wykonanie strony graficznej jest bardzo dobre. Owa mroczność widoczna jest właśnie w dobrych i czytelnych teksturach. Digitalizowane postacie, nieźle wykonany widok broni (w tym naszych stóp i pięści), ładnie prezentujące się efekty specjalne, wybuchy, przezroczystość, nawet widok mapy jest nietuzinkowy. Dobra ilość klatek animacji.



Program nie ma jakichś wygórowanych wymagań sprzętowych. Demo ruszy już na Amidze 020 z 4 MB Fastu i AGA (w trybie 2x2 piksel). A mając 030, można pokusić się o rozrywkę w lepszych detalach. A wszystko to z zadowalającą prędkością. Docelowo „Brain Killer” obsługiwać miał także karty graficzne.

W kwestii dźwięku nie ma za dużych rewelacji. Choć podejrzewam, że gdybym opisywał pełną wersję, to noty byłyby znacznie wyższe. Ale do rzeczy. Podczas przedzierania się przez labirynty, towarzyszy nam dość poprawnie zrobiony moduł. Jest przyjemny dla ucha, nie agresywny, ale też nie przesłodzony. Dobrze komponuje się z akcją, podtrzymując klimat. Co do wszelkich sampli i efektów dźwiękowych,

to są raczej średniej jakości. Kroki, strzały, wybuchy, okrzyki naszych wrogów itp. są dość poprawnie wykonane, ale nie szokują jakimiś rewelacjami.



Strzelanka 3D, połączona z elementami bijatyki to niewątpliwie ciekawy pomysł na wykonanie FPS'a. Niezła grafika, zjadliwe udźwiękowienie, nie za wielkie wymagania sprzętowe, to cechy, dzięki którym „Brain Killer” znalazłby amatorów, gdyby ukazał się na rynku. Ostatecznie rzecz biorąc, jeżeli gra by jednak ujrzała światło dzienne (może jakiś mały apel do twórców?), to z chęcią bym w nią zagrał, a potem jeszcze raz opisał.

Gatunek – FPS s-f, z elementami bijatyki

Producent – Virtual Madness

Wydawca – Titan – Worldwide

Rok – 1997 (rok ukazania się demo)

Ilość dysków – program ma około 4 MB, HD

Platforma – 020, AGA, karta GFX

OCENA:

Grafika – 90

Dźwięk – 65

Grywalność – 85

Ogółem: 80

Testament, Testament 2

Kiedy człowiekowi zaczęło być za dobrze na Ziemi, z nudów i przesycenia szczęśliwością, zaczął wymyślać historie, w których stawiał siebie w obliczu zagrożenia, bynajmniej nie pochodzącego od innego człowieka. Takim to sposobem doszło do powstawania najróżniejszych bajek, legend i mitów, w których to ród ludzki nękany był przez najróżniejsze stworzenia, nie pochodzące z tego świata. A to, czy te opowieści, na których bazują dziś pisarze, scenarzyści i twórcy gier są rzeczywiście zmyślane, czy też mają swoje realne podstawy, to chyba wiedzą tylko nasi praprzodkowie. Nam pozostaje wierzyć lub nie, a już na pewno możemy sobie zaserwować grę opartą na jakiejś ciekawej, mrożącej krew w żyłach, pełnej zjawisk paranormalnych fabule.

Po tym krótkim wstępie zapraszam do meritum sprawy. Przedstawione poniżej tytuły to FPS'y, których fabuła opiera się właśnie na zjawiskach żywcem wyjętych z dreszczowców o duchach, zjawach, żywych trupach czy wyznawcach szatana. Jednym słowem, cała śmietanka towarzyska horror-show biznesu. Biorąc pod uwagę fakt, że obie gry zbytnio nie różnią się od siebie opisywaną historią, nie będę specjalnie rozgraniczał opisu na pierwszą i drugą część.



Cel gry raczej nie odbiega od jakichś przyjętych norm: powystrełać wszystkich przedstawicieli świata nadprzyrodzonego oraz znaleźć wyjście. Czasami, błąkając się po niezbyt przyjemnych lokalizacjach, takich jak cmentarzyska, katakumby, labirynty czy też nawiedzone zamczyska, musimy poszukać odpowiednich kluczy, pasujących tylko do jednego rodzaju drzwi. Czyhające na nas na

każdym kroku chordey wyżej wspomnianych pomyłków zmuszają nas do częstego sięgania po niezbyt okazały arsenał (m.in. pistolet, karabin, miotacz ognia czy rakieta). Po drodze uraczyć możemy się skromnie rozstawionymi apteczkami, a także zasilić broń w dodatkową amunicję.

Powrót pikselozy. Tak jednym stwierdzeniem można by było scharakteryzować stronę graficzną przedstawianych propozycji. Niepełne 3D (tylko jedna płaszczyzna i żadnych schodów) pokryte jest średnio wykonanymi teksturami, które swoją dość ciekawą kolorystyką próbują nadrobić niedociągnięcia, budując odpowiedni klimat grozy podczas rozgrywki. Wszelakiego rodzaju obiekty, oponenty czy inne elementy scenografii, wykonane tylko i wyłącznie z bitmap, dają nam do zrozumienia, że prócz szkieletu nie zobaczymy już żadnych innych obiektów trójwymiarowych.

Tak ogólnie można przedstawić grafikę w obu pozycjach, chociaż w dwójce nieco poprawiono te wady. Rozpikselowanie jest nieco mniejsze, wygładzono troszkę widok naszych wrogów, dodano efekt przezroczystości (np. duchy) oraz ruchome niebo, nowe postacie i poziomy. Pozostałość jest podobna. Gorzej ma się niestety animacja w grze i nie chodzi o ogólną płynność (bo ta jest dość dobra), gdyż programy docelowo dedykowane były dla po-





siadaczy słabszych konfiguracji (gra ruszy na gołej Agacie, a do pełniejszego komfortu wystarczy tylko 4 MB Fastu i twardziel, ewentualnie zaleca się także 030), lecz o liczbę klatek. Postacie poruszają się dość sztywno i skokowo.

Jeszcze parę słów o konfiguracji ekranu. Znane już rozwiązania umożliwiają nam zmianę takich parametrów, jak rozmiar okna, szczegółowość wyświetlanych tekstur czy

włączanie i wyłączanie nakładania grafiki i dodatkowych efektów graficznych.

Duży wkład w stworzenie owego mrocznego klimatu w obu częściach ma strona dźwiękowa. Mimo, że sample czy moduł nie grzeszą doskonałością, a ich wykonanie przesyczone jest momentami słyszalnym syntetyzmem, to wszelkie odgłosy dobiegające z zaświatów potrafią w odpowiednich momentach zjeżyć włos

na głowie. Przeróżne krzyki czy też sakralnie brzmiące „luuu” potrafią zadowolić niejednego fana horrorów. Pozostałość efektów dźwiękowych, strzałów z broni, otwierających się drzwi czy odgłos kroków są już raczej standardowe

dla większości podobnie wykonanych FPS'ów.

Pomijając fakt, że przedstawione tytuły nie odniosły jakichś przerażających sukcesów, a ich wykonanie sprawia wrażenie cofnięcia się w technicznym rozwoju, to panujący klimat dobrze wpływa na grywalność. „Testamenty” będą dobrą pozycją nie tylko dla amatorów dreszczowców i horrorów, ale także dla wszystkich fanów FPS'ów lubiących się troszkę pobać. A gwarantuję, że rozgrywka prowadzona w środku nocy, przy zgaszonym świetle przysporzy nas o przyspieszone bicie serca.



Gatunek – FPS horror

Producent – Insanity

Wydawca – APC & TCP (Testament), Magazyn Amiga – Polska (Testament 2 w wersji shareware)

Rok – 1997 (Testament), 1998 (Testament 2)

Ilość dysków – 3, HD (Testament), program ma około 4 MB, HD (Testament 2)

Platforma – 020, AGA

OCENA:

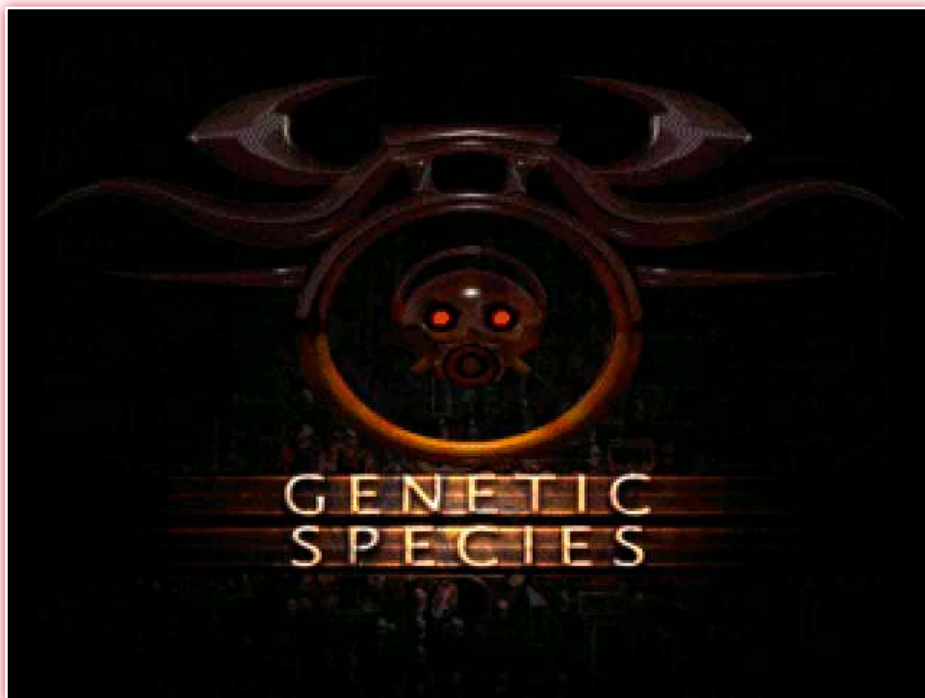
Grafika – 65

Dźwięk – 75

Grywalność – 90

Ogółem: 75

Genetic Species



O tej grze mawiało się swego czasu: „główny konkurent amigowego Quake’a” (no, ale tym drugim zajmiemy dopiero się w następnym opisie). Rozwój amigowych FPS’ów stanął w takim momencie, że programiści przestali już patrzeć wstecz, tworząc kolejne tytuły tak, aby można było (jako tako przynajmniej) zagrać w tę czy tamtą grę na podstawowym, nierozbudowanym sprzęcie. Choć teoretycznie minimalne wymagania opisywanego produktu to właśnie podstawowa 020 z AGĄ (o 8 MB Fastu, napędzie CD i twardej nie wspominając), to komfort grania na takiej nierozbudowanej (przynajmniej o kartę turbo) Amidze nie był nawet jako taki. A dla amatorów porządných i coraz lepszych strzelanek 3D (i nie tylko strzelanek 3D) nadszedł czas rozbudowy amigowego hardware’u.

Po tym, nieco sprowadzającym na ziemskie realia (czy raczej amigowe) wstępie, przejdźmy do właściwego opisu.

Pod koniec XXI wieku badaniami naukowymi i różnymi przedsięwzięciami na skalę nie tylko globalną, ale także sięgającą poza granice atmosfery ziemskiej, zajmowały się korporacje. Ich, czasami, niesmaczna rywalizacja, doprowadzała do konfliktów zbrojnych. Chroniąc swoich interesów i tajemnic, wszczęto pracownikom mikrochipy, wymuszające

na nich lojalność i posłuszeństwo. Badania, jakie prowadzono poza granicami Ziemi, często przekraczały normy etyczne i moralne.

Rządzące i toczące boje między sobą korporacje swoją rosnącą potęgą militarną zaczynały zagrażać globalnemu bezpieczeństwu. Powołana do życia organizacja Sprzymierzonych Sił Oporu ma na celu pokrzyżować prawdo-

podobnie destrukcyjne zamiary wobec mieszkańców Ziemi, planowane przez korporacje. I choć wydawało się, że młoda organizacja nie jest jeszcze na tyle silna, aby przeciwstawić się niszczycielskim gigantom, to wydarzenia z roku 2208 wpłynęły znacząco na dalszy bieg wydarzeń.

Podczas walk z udziałem nieznaną nam pastników, dwie spośród trzech baz, znajdujących się na Księżycu zostają zmiecione z powierzchni Srebrnego Globu. Podejrzewa się, że ma to związek z asteroidą Atlas, krążącą wokół Księżyca, gdzie dawniej była kopalnia górnicza, obecnie przekształconą w bazę wojskową. Nasze zadanie to wysłanie jednostki bojowej o nazwie Bioshifter do jedynej ocalałej bazy załadunkowej, wybudowanej przez korporację Cantex w celu rozeznania w sytuacji.

Podczas lotu statek transportowy zostaje zaatakowany przez niezidentyfikowanych pastników, co nieco komplikuje sprawę, stawiając coraz to nowsze niewiadome. Walka w przestrzeni kosmicznej pochłania dwóch spośród trzech pilotów eskortujących transportowiec. Ale ich życie nie idzie na marne. Oponenti zostają zestrzeleni, a nasz załadowek dociera do celu.

Po tej wprowadzającej historyjce, połączony ze świetnym, renderowanym intrem, zaczy-





na się właściwa rozgrywka. Jako wspomniana samobieżna maszyna bojowa Bioshifter, wkraczamy do kompleksu bazy, aby sprawdzić, o co w tym wszystkim chodzi. Ale zamiast prowadzić badania i rozeznanie, od razu zostajemy przywitani przez uzbrojonych żołnierzy w skafandrach, którzy ani myślą prowadzić z nami wyjaśniające rozmowy. Notabene, ich owe dość wrogie nastawienie częściowo wyjaśnia sytuację. Cóż, taki los maniaka strzelanek 3D. Znowu trzeba walić do wszystkiego, co się

rusza. Ale nasi wrogowie to nie tylko ludzko wyglądające postacie, pracownicy stacji, inżynierowie, robotnicy. W miarę przechodzenia do kolejnych poziomów, na naszej drodze stawać będą osobniki niewiele mające wspólnego z człowiekiem. Do „naszej dyspozycji” są także różnego rodzaju roboty, cyborgi, żywe trupy oraz jakieś przebrzydłe, zmutowane robale i pajęczaki. Te ostatnie są zapewne wynikiem wspomnianych wcześniej, mijających się z zasadami etyki, genetycznych doświadczeń. Arse-

nał na samym początku nie powala na kolana. Wchodząc do Cantex'u, wyposażeni jesteśmy w jakąś standardową spluwę. Ale im dalej się znajdujemy, tym bardziej możemy się obładować (maksymalnie cztery sztuki) najróżniejszym żelastwem: m.in. rakiетnicą, pistoletem plazmowym, karabinem, granatami, miotaczem ognia, laserem, a nawet toporem strażackim, przydającym się do walki wręcz. Ciekawostką jest to, że nie na każdym poziomie można użyć wszystkich broni (pod warunkiem ich posiadania). Choćby ze względu na bezpieczeństwo kompleksu (np. strzał w szybę z karabinu może spowodować, że nagle zostaniemy wysłani w przestrzeń kosmiczną, z powodu próżni rzecz jasna). Poza tym całym strzelaniem, grając w GS trudnimy się także zbieractwem. Począwszy od amunicji do naszych zabawek, poprzez apteczki reperujące uszkodzenia w Bioshifterze, na różnych gadżetach typu karty kodowe do drzwi, dyski zawierające poszukiwane informacje, a także biotoksyny, kończąc. Zbieranie owych gadżetów łączy się czasami z dodatkowymi zadaniami na poszczególnych etapach, o których to zostaniemy poinformowaniu przed rozpoczęciem poziomu. Poza tym, podczas gry możemy korzystać z terminali, w których to uzyskujemy dostęp do menu głównego, w którym mamy możliwość zmodyfikować ustawienia (a jest w czym wybierać: od konfiguracji ekranu, szczegółowości grafiki, poprzez opcje językowe, poziomu trudności, na konfiguracji sterowania i dźwięku kończąc), zapisać lub wczytać stan gry itp.



Może nietypowo, ale zacznę od wytykania błędów. Pierwszym rzucającym się w oczy, po opadnięciu wszystkich „ochów i achów”, defektem w stronie graficznej jest brak pełnego trzeciego wymiaru. Cała akcja toczy się tylko na jednej płaszczyźnie. To niewątpliwie umniejsza grze prestiżu. Druga sprawa to głębia ekranu. Maksimum to 8 bitów, czyli 256 kolorów, bez względu na to, czy mamy tylko AGĘ, czy też zaopatrzyliśmy się w kartę graficzną. Może to nieco rozczarować posiadaczy tego cennego nabytku. Wynagrodzeniem tego niedociągnięcia może być fakt, że posiadacze kart GFX mogą pograć sobie w rozdzielczości 640x480, podczas gdy na samej ADZE maksymalnie można uzyskać 320x250. Mimo specjalnych procedur wygładzających piksele, czasami można zauważyć niestety kostkowaną grafikę, szczególnie wtedy, gdy zbyt blisko podejmiemy do naszych wrogów.

Brak pełnego 3D rekompensuje fakt dobrze zaprojektowanych poziomów. Są one rozbudowane, z dużą ilością korytarzy i labiryntów. Duża ilość pomieszczeń, pokoi, śluz, różnych elementów scenografii potrafi dać wrażenie pełnej trójwymiarowości. Dopracowane tekstury oraz wspomniane wyżej procedury wygładzające świetnie budują wizerunek wirtualnego świata. W zachwyt mogą także wprowadzić niektóre efekty: zabawy światłem, flara (np. podczas strzału z plazmy, lecący pocisk oświetla ciemny korytarz na całej długości), przezroczystość niektórych elementów (szyby, blokady laserowe), wybuchy, czy nawet widok gorącego, falującego powietrza. Całość dobrze wykonana i bez większych zastrzeżeń. Warto także wspomnieć o wspomnianym wcześniej intrze. Renderowana



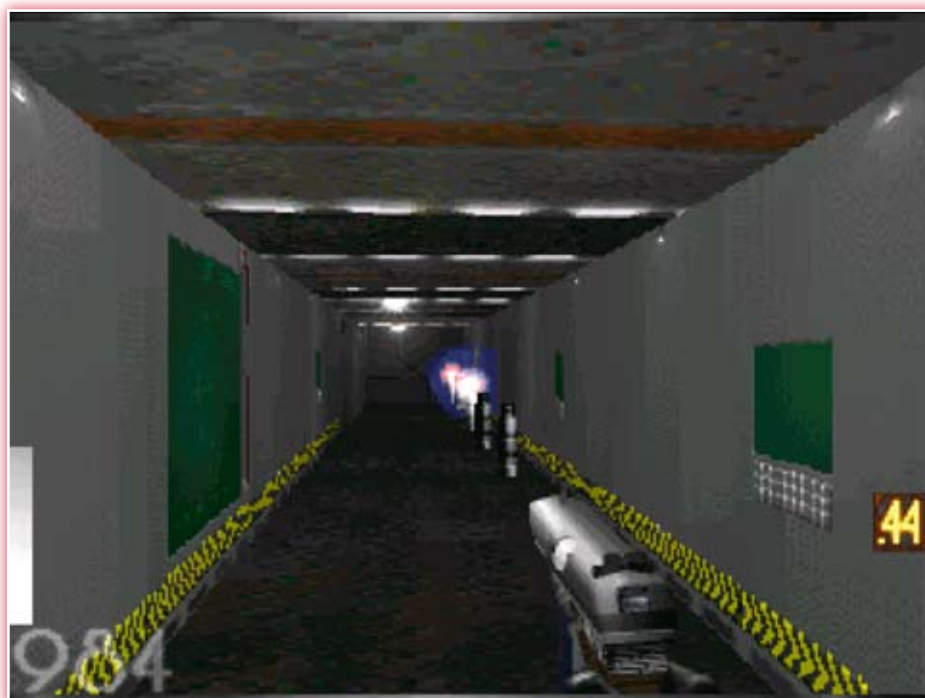
animacja (około 200 MB!) jest tak zrobiona, że bardziej przypomina krótki filmik 3D, niż proste wprowadzające demo. Dzięki takiemu wykonaniu od razu narzuca ona pewien klimat i charakter gry. Oczywiście na korzyść.

Jak do tej pory nie spotkaliśmy się z pozycją, która korzystałaby z pakietu AHI. Dzięki temu nie tylko nasza PAULA dostaje kopa, ale możemy także skorzystać z kart muzycznych. Nawiasem mówiąc, poziom udźwiękowienia nie byłby możliwy bez zaimplementowania obsługi owego pakietu. I tak, w GS nie usłyszymy już syntetycznych sampli czy protraktersko brzmiących modułów. Strzały z broni, przeładowanie magazynku, krzyki wrogów, głos komputera pokładowego, otwieranie drzwi czy

nawet huczący wentylator sprawiają wrażenie realizmu. Na dodatek ścieżki muzyczne dopełniające kompakt dobrze się prezentują, utrzymując klimat w grze.

„Genetic Species” to gra, która zabrała mi z życiorysu wiele nocnych godzin. Niezła strona graficzna, realistyczne udźwiękowanie, a przede wszystkim rozwijająca się fabuła (mowa tu o wspomnianym infie, ukazującym się przed rozpoczęciem poziomu, które także odkrywa przed nami kolejne fragmenty tajemnicy spowijającej bazę Cantex) – wszystko to może doprowadzić do stanu, kiedy całkowicie zatracimy się w grze. Choć w prezentowanej pozycji można doszukać się jeszcze wielu niedociągnięć, to niewątpliwie jest to tytuł lokujący się w czołówce amigowych FPS’ów. Polecam tę pozycję amatorom trójwymiarowego strzelania połączonego z klimatem thrillerów s-f.

Na koniec warto dodać, że autorzy wypuścili do gry kilka łatek i ulepszeń, dodając między innymi opcję multiplayer oraz rozgrywki przez sieć.



Gatunek – FPS s-f
Producent – Marble Eyes
Wydawca – Vulcan Software
Rok – 1998
Ilość dysków – 1 CD, HD
Platforma – 020, AGA, karta GFX
OCENA:
 Grafika – 90
 Dźwięk – 95
 Grywalność – 85

Ogółem: 90

Quake

Tej pozycji raczej nie trzeba nikomu specjalnie przedstawiać. Fenomenem tej gry jest to, że jest ona nie tylko znana na całym świecie, ale stworzyła także pewnego rodzaju subkulturę. Spotkałem się nawet z przykładem wprowadzenia tytułu do mowy potocznej: jeśli ktoś chce kogoś obrazić, mówi mu wtedy: „ty kłejku!”.

A tak na serio, pojawienie się „Quake’a” przetrzało nowe szlaki w dziedzinie FPSów. Wyzaczyło nowe standardy technologiczne. Sukces medialny tej gry polegał nie tylko na wprowadzeniu nowych rozwiązań w sferze silnika graficznego, efektów dźwiękowych, fabuły czy możliwości rozgrywki przez sieć, ale także dzięki konwersji na jak największą ilość platform. Firma ID Software wypuszczając swój produkt, myślała przede wszystkim o rynku PC. Szybko się jednak okazało, że się myliła. Click Boom podjęła się jej konwersji i dzięki temu „Quake” ukazał się także na Ami klasyczną. I choć jest ona wierną kopią wersji pecetowej, to jej kod został przepisany od podstaw z jednoczesną optymalizacją pod kątem procesorów mk68. Ale to nie wszystko. Dwójka (która ukazała się także na konsolki) i kolejne odsłony, dzięki portom (o których na końcu artykułu) trafiły także pod dachy użytkowników AmigPPC, Amig z OS 4.x oraz platform amigopodobnych: MorphOS i AROS. No, ale wracając do meritum sprawy, pomimo faktu, że po tytuł sięgnął niemalże każdy amator FPSów, pozwolę sobie jednak przytoczyć szczegóły jej scenariusza.

Wcielając się w rolę śmiałka naszym zadaniem jest pokrzyżować złowieszcze plany bestii o imieniu Shub-Niggurath. Aby dokonać tego czynu, musimy przebrnąć w całym kawałku przez cztery epizody, zbierając fragmenty magicznego artefaktu, którego poskładanie umożliwi nam spotkanie się z naszym czarnym charakterem. Ale cwany Shub, spodziewając się naszej niechcianej wizyty, wysłał przeciwko nam całe zastępy swoich krwiożerczych sług. Na swojej drodze spotkać będziemy mogli m.in. całe chordy atakujących stadami potworów, rycerzy, lewitujących w powietrzu duchów, zombie rzucających w nas fragmentami swojego gnijącego ciała, czy inne piekielne monstra. Na nasze życie będą także czyhać większe gabarytowo stworzenia oraz bossowie. Niektórzy z nich są uzbrojeni w jakąś „ludzką” broń, np. rakiety albo granaty. Ich ogromne zróżnicowanie pod względem wyglądu, form ataku, zachowania (sztucznej inte-

ligencji) naprawdę dostarczają wstrząsających wrażeń. A w obliczu faktu posiadania tylko jednej kuli w magazynku i korytarza wypełnionego po brzegi złowieszczymi stworami, możemy zostać zmuszeniu do ucieczki gdzie pieprz rośnie. A gdy już uda nam się na chwilę schronić w pozornie bezpiecznym miejscu, zauważymy jak ze strachu drżą nam dłonie, nerwowo spoczywające na klawiaturze. Co do kul, nasz zestaw broni również jest imponujący. Do zabawy z naszymi ziomkami mamy osiem różnych gnatów. Każdy z nich charakteryzuje się inną amunicją, zasięgiem oraz siłą rażenia. Do walki wręcz posłużyć nam może topór. Jego zaletą jest to, że nie potrzebuje amunicji, ale jest także najsłabszą bronią. Kolejne to dwa rodzaje strzelb: jedno- i dwulufowa. Dość ekskluzywnym modelem jest Nail-Gun. Choć zachowuje się jak karabin, to z jego pomocą możemy sygnąć w twarz potworowi garścią gwoździ. Zapodanie z rakiety czy granatnika przydaje się podczas zmasowanych ataków. Ale niestety, nieumiejętne używanie tych modeli (np. gdy granat lub raketa wybuchnie zbyt blisko nas) może doprowadzić do tego, że odstrzelimy sobie nogę. Ostatnim na liście jest miotacz impulsów elektrycznych o wysokim napięciu. Broń niezwykle skuteczna, potrafiąca w przeciągu jednego, dwóch strzałów powalić kilka sztuk, ale wysoce niewskazana podczas przebywania w wodzie. Użycie jej podczas babrania się w bajorku może doprowadzić do niewskazanej defibrylacji (w tym przypadku: śmiertelności porażenia prądem).

Prócz samej walki z oponentami i szukania kolejnych elementów artefaktu, po drodze zbieramy wiele przydatnych gadżetów. Są to wspomniane modele broni, amunicja do nich, apteczki, pancerze, skafandry do wędrowania w kwasie oraz zestawy do nurkowania. Podczas przeczesywania poziomu warto poszukać tzw. sekretnych miejsc, gdzie owe gadżety tylko czekają, aby je zebrać. Główną metodą w przemieszczaniu się jest szukanie przełączników odpalających windy, otwierających drzwi czy opuszczających mosty.

Wspomniane we wstępie nowe standardy wyznaczone przez „Quake’a” mają się tu przede wszystkim do rozwiązań w sferze wykonania grafiki. I choć wśród amigowych FPSów znalazły się wcześniej opisywane pozycje, charakteryzujące się pełną trójwymiarowością, to – w przeciwieństwie do tej gry – ich engine 3D nie zawierał aż tylu szczegółów. Korytarze, wielkie i małe pomieszczenia, schody, półki, półpiętra, windy, skwerki pod gołym niebem, podwodne komnaty, wrogowie, wszelkie elementy scenografii, a nawet niektóre pociski, są wykonane z wektorowych brył pokrytych szczegółowo wykonanymi teksturami. Nie ma mowy o jakimś rażącym rozpikselowaniu.

Jak przystało na klimat lochów i zamczysk opanowanych przez złe moce, paleta jest mroczna, lekko ciemnawa, z dominującą przewagą odcieni brązów i zgniłej zieleni. Do tego wszelkie efekty specjalne, zabawy światłem,



efekt flary, przezroczystości i krwistych scen, nawet falowanie obrazu podczas przebywania pod powierzchnią wody. Wszystko perfekcyjnie wykonane. Jedynym minusem mogą być tu wybuchy, które zamiast wyglądać jak wybuchy (tak jak w niektórych poprzednich pozycjach), wyglądają jak rozpadające się na cztery strony świata czerwono-żółte kropki lub kwadraciki. Mimo dość wystarczających możliwości konfiguracyjnych, zmiany rozdzielczości ekranu czy wielkości okna, płynność gry będzie zależała od sprzętu, jakim dysponujemy. I choć minimalna konfiguracja podawana przez producenta to procesor 020, kości AGA, 8 MB Fastu (a także napęd CD i twardego dysku), animacja na takim sprzęcie, nawet na najniższych ustawieniach, będzie przypominała slideshow, z prędkością rzędu dwóch, trzech klatek na sekundę. Na jako taką grę można sobie pozwolić wyposażając Amię w kartę z prociem 030/50 i 16 MB RAMu. Ale do pełni szczęścia zaleca się procesor min. 040/40 i kartę graficzną.

Ścieżkę muzyczną do „Quake’a” napisał lider zespołu „Nine Inch Nails” – Trent Reznor. Jej klaustrofobiczno – psychodeliczny charakter bardzo dobrze wpływa na klimat grozy, a w połączeniu ze świetnie wykonanymi efektami dźwiękowymi i wszelkimi odgłosami daje naprawdę ciekawe odczucia. W amigowej wersji, na szczęście niczego tu nie zmieniano. A dzięki implementacji pakietu AHL, nie tylko PAULA nabiera obrotów, ale można także wykorzystać karty muzyczne wpięte w przyjaciółkę.

Biorąc pod ostrze grywalność, nie trzeba się zbytnio rozpisywać. Wszystkie wymienione zalety mówią same za siebie. No tak, ale nie ma dymu bez ognia. I dla Amigi nastały czasy, że bez odpowiedniej rozbudowy nie ma co marzyć o sięganiu po coraz nowszy soft. I choć koło „Quake’a” na pewno przejdą obojętnie amatorzy retro sprzętu, dla których Amia kojarzy się przede wszystkim z nostalgią i pierwszą częścią „The Settlers”, to fani coraz nowszych, coraz lepszych i coraz bardziej wymagających tytułów i programów dobrze wiedzą, że dystans, jaki dzieli klasyka od dzisiejszych standardów, można zmniejszyć tylko poprzez jego rozbudowę.

W zasadzie to mógłbym już zakończyć nasz cykl, podsumowując go smutnym akcentem, gdyż „Quake” był ostatnim napisanym i wydanym na klasyka mk68 FPS’em. Ale tego nie zrobię, ponieważ na sam deser zostawiłem porty. Co prawda, skupię się tutaj może nie na samych opisach pecetowych gier, przeportowanych na przyjaciółkę, ale na przedstawieniu samego zjawiska.

No dobra, to w takim razie trzeba by było się zastanowić, czym jest port. Najogólniej mówiąc, jest to kod pliku wykonywalnego oryginalnego programu lub gry, przepisany na inny procesor lub system. Tak przerobiony program korzysta z tych samych zasobów (bibliotek, plików z grafiką czy muzyką) co oryginał, ale uruchamia się na innej platformie.



Za początek tego zjawiska przyjmuje się rok 1998. Wtedy to, po upublicznieniu przez ID Software kodu źródłowego, dosłownie na dniach powstały pierwsze amigowe porty „Doom’a”. W następnej kolejności były „Hexen” i „Heretic”.



Były to aplikacje darmowe, które do pracy wymagały oryginalnej wersji pecetowej (komercyjnej lub shareware). Pisanie ich całkowicie pod system umożliwiało korzystanie z dodatkowego hardware’u wpiętego w przyjaciółkę, począwszy od kart turbo, poprzez AGE, kartę graficzną, na karcie muzycznej kończąc. Takie rozwiązanie dawało programistom duże możliwości, ale kosztem użytkowników słabszego sprzętu. Wspomniane wyżej pierwsze trzy porty do jako takiej pracy wymagały już AGI, procka 030 i 8 MB Fastu. A swoje skrzydła rozwijały dopiero przy 040 i karcie graficznej.

Kolejnymi portami pecetowych FPS’ów, które ukazały się na Ami mk68, był ojciec wszystkich trójwymiarowych strzelanek, „Wolfenstein 3D” i druga część „Doom’a” (oba z wymaganiami podobnymi do poprzedników) oraz „Duke Nukem 3D” (przy czym ten ostatni, bezwzględnie już wymagał 060 i bardzo dużo Fastu).



Rozwój dwuprocessorowych kart turbo, opartych na MC68 i PPC, produkowanych przez Phase5, pozwolił twórcom portów na przenoszenie na Amigę coraz lepszych i bardziej zaawansowanych tytułów z wykorzystaniem drżemiej w powerkach mocy. Dzięki temu klasykowcy doczekali się jeszcze dwóch komercyjnie wydanych, opartych właśnie na amigowych portach, FPS’ów: „Heretic II” (Hyperion/Titan, rok 2000) oraz „Quake 2” (Hyperion, 2002).



Portowanie programów i gier (nie tylko FPS’ów) to w dzisiejszych czasach główny sposób na to, aby kolejne dobre tytuły z innych komputerów ukazywały się także na naszej przyjaciółce. I choć sprzętowo obecnie w grę wchodzi raczej następca Amigi Classic (A1, MorphOS czy AROS), to w sieci można od czasu do czasu spotkać jeszcze aplikację, dzięki której posiadacze Amigi mk68 (w rozbudowanych zestawieniach) będą sobie mogli zapodać jakiś ciekawy tytuł. Co prawda fajnie by było, gdyby na Amię klasyczną czy nową od czasu do czasu wydawany był jeszcze jakiś dobrze zrobiony FPS, nie będący jednocześnie portem czy konwersją, to jednakże dzięki temu zjawisku nasze komputerki mogą ciągle brać czynny udział w tej, swego rodzaju, globalizacji oprogramowania.

Gatunek – FPS horror/s-f
Producent – Click Boom (PC – ID Software, 1996)
Wydawca – PXL
Rok – 1998
Ilość dysków – 1 CD, HD
Platforma – 020, AGA, karta GFX
OCENA:
 Grafika – 95
 Dźwięk – 95
 Grywalność – 95

Ogółem: 95

Don Rafito

Pograjmy jak za dawnych lat...

ROK 1984 - cz. 1

Gdyby ktoś na chybił-trafił porównał pierwszy lepszy tytuł wydany w 1984 roku z jakąś pozycją z roku poprzedniego, pochopnie mógłby dojść do wniosku, że niewiele w produkcji gier się zmieniło. Niemilosierne kanciasta grafika, piskliwy dźwięk i mało oryginalne tytuły, gatunkowo ściśle przynależące do obszaru o granicach wyznaczonych przez „Donkey Kong” i „Manic Miner”, to mocne argumenty na obronę tej tezy. Kolejnym, przemawiającym za tym, iż firmy stawiały raczej na ilość, aniżeli na jakość, jest fakt, iż to właśnie w 1984 roku powstała największa liczba gier w całej historii C-64. Pozornie trudno odeprzeć te argumenty. Jednakże tak naprawdę w produkcji gier zmieniło się całkiem sporo. I chociaż w dalszym ciągu intensywnie posilkowano się konwersjami z ZX Spectrum czy automatów, chociaż nadal powstawała ogromna ilość klonów różnorodnych hitów, to jednak – po pierwsze – pojawiało się coraz więcej oryginalnych pomysłów. Po drugie – coraz częściej można było spotkać gry wykonane z niespotykaną dotąd starannością. Po trzecie – gatunki, które w 1983 roku były reprezentowane przez pojedyncze, pionierskie produkcje, w roku Orwellovskim zaprezentowały pełnię swoich możliwości. Odnoszę też wrażenie, iż coraz lepiej wykorzystywano niesekwencyjny dostęp do danych, jakie oferowały stacje dysków, dzięki czemu powstały gry o niespotykanym dotąd rozmachu. Tyle w ramach wstępu, pora przejść do konkretów, których tym razem będzie wyjątkowo dużo. Dlatego postanowiłem podzielić artykuł na (co najmniej) trzy części. Rozpocznijmy od gier najprostszych, które równie dobrze mogły powstać w 1983 roku.

Przyjrzyjmy się najpierw produkcji, która była swego rodzaju holdem – a zarazem podsumowaniem – pierwszej ery gier komputerowych. Mówię o epoce, której symbolami były takie tytuły jak „Space Invaders” bądź „Frogger”. Pomysł na napisaną przez Davida Whittaker’a, a wydaną przez Terminal Software **Lazy Jones** był prosty – zebrać różne prościuteńkie gry w jednej produkcji i połączyć je jakimś scenariuszem, który by taką zbieraninę uzasadnił.

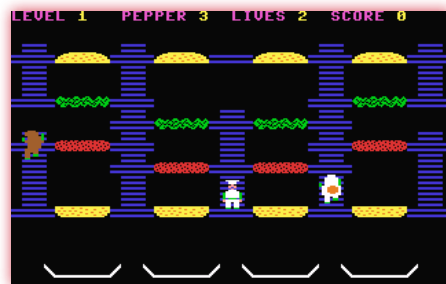
W „Lazy Jones” wcielamy się w rolę tytułowego lenia, który ani myśli wykonywać swoje obowiązki (jest sprzątaczem). Zamiast tego



woli – kompletnie mu się nie dziwi – doskonalić swoje umiejętności w grach video. Bohater porusza się po trzech kondygnacjach, między którymi przemieszcza się za pomocą windy. Musi uważać na swojego szefa, ducha byłego kierownika (ciemna sylwetka) i ruchomy wózek. Na każdym piętrze znajduje się sześć sal. Z tej osiemnastki większość zawiera automaty do gier, w czterech pozostałych znajdują się: schowek na miotły, widok których wywołuje u bohatera palpitacje serca, łóżko, gdzie bohater śni koszmara o swoim szefie, ubikacja i bar. Ta ostatnia lokalizacja to również coś w rodzaju „gry w grze”, w której **Lazy Jones** musi przeskakiwać nad pijakiem i częstować się alkoholem serwowanym przez barmana. I chociaż sam pomysł cieszy moje serce gracza, to jednak nie jestem fanem tej produkcji. Powód? Cóż, gry, z którymi bohater może się zmierzyć, są skrajnie prościutkie i – na dłuższą metę – po prostu nudne. Każda z nich toczy się do straty jedyne go życia, bądź do upływu czasu danego na rozgrywkę. Oprócz wyżej wspomnianych mamy tu m.in. klon takich gier jak „Snake”, „Chuckie Egg” bądź „Breakout”. Warto też wspomnieć o ścieżce dźwiękowej. Chociaż dzięki ciągłemu umpa-umpa wydaje się, iż słyszymy jedną melodię, to tak naprawdę zmienia się w zależności od pomieszczenia, w którym przebywamy. Tym samym mamy okazję, by posłuchać m.in. SID’owskiej wersji „99 Luftballons”. Inny z tych mini utworów został zremiksowany przez niemiecką grupę Zombie Nation a ich kawałek, „Kernkraft 400” stał się hitem. Nie obyło się bez prawnych przepychanek zakończonych zapłaceniem tantiem Davidowi Whittaker’owi.

Jednoekranowe platformówki to gatunek, który wkrótce miał przejść do lamusa. Gier w tym stylu powstało w 1984 roku mnóstwo i chociaż nie posuwały one historii rozwoju gier

komputerowych ani o milimetr do przodu, kilka z nich warto jednak w tym miejscu przypomnieć. Na dobry początek hit rodem z automatów – **Burger Time**. Powstała w 1982 gra, doczekała się konwersji na C-64 (wydanej przez Interceptor Software) właśnie w 1984 roku.



Idea jest prosta. Bohater – kucharz Peter Pepper – chodzi sobie po różnych platformach, na których zostały umieszczone fragmenty hamburgera: dwa kawałki bułki, szynka i sałata. Zadaniem szefa kuchni jest przejście po każdym ze składników tak, by pod platformami, na przygotowanych talerzach, uformował się pyszny (?) hamburger. Jego przeciwnicy również przynależą do kategorii: gastronomia. Kucharz musi uważać na Pana Hot Doga i Pana Jajko, których albo może zatrzymać jedną ze skromnie wydzielonych porcji pieprzu, albo też zgnieść fragmentem burgera. Oryginalny pomysł i zabawne wykonanie powodują, iż gra – pomimo skromnych wartości estetycznych – potrafi wciągnąć. Tutaj jednak należy ostrzec, iż zadanie postawione przed graczem do prostych nie należy – po każdej stracie życia formowanie hamburgerów trzeba rozpocząć od nowa, a poziom trudności rośnie dość szybko. „**Burger Time**” doczekał się sporej liczby konwersji i kontynuacji na różnych platformach, w tym również na telefonach komórkowych. Jeśli natomiast o C-64 idzie, to w tym samym roku ukazała się gra pt. „**Burger Chase**”, natomiast w 1997 – „**Burger Time’97**”. Jest i polski akcent – swoją wersję gry (o tytule „Cheeseburger”) napisał przed laty Krystian Grzenkowicz. Z kolei wydawca „**Burger Time**”, Interceptor Software, w tym samym 1984 roku popisał się ciekawą grą Iana Graya, łączącą elementy platformówek ze specyficzną strzelanką – „**Tales of the Arabian Nights**”. Ten tytuł omawiam w innym miejscu tego numeru.

Dlatego przejdę od razu do innego hitu. **Henry's House** to jedna z najładniej zaprojektowanych jednoekranowych platformówek. Wydana przez English Software gra (autorstwa Chrisa Murraya) składa się jedynie z ośmiu poziomów, jednakże jej „legalne” ukończenie (czytaj: bez cheatów) nie należy do łatwych zadań.



Bohaterem jest mały Henry, który nierozważnie skosztował mały łyk specyfiku znalezionej w laboratorium swojego ojca - króla. Tato bohatera musiał być wielkim fanem „Alicji w krainie czarów”, gdyż Henry skurczył się do wielkości 6 cali i na dodatek utknął w szafie. Jego zadaniem jest przeprawienie się przez wszystkie pomieszczenia tytułowego domu, by ostatecznie znaleźć lekarstwo na swoją dolegliwość. Aby to osiągnąć, musi posprzątać wszystkie te pokoje, gdyż klucze do drzwi gdzieś zawieruszyły się w ogólnym bałaganie. Tym samym „**Henry's House**” to w gruncie rzeczy specyficzna wariacja na temat „**Manic Miner**”, w której po zebraniu wszystkich rozrzuconych przedmiotów, dzięki odnalezionemu kluczowi, można przejść do kolejnego poziomu. W porządkach przeszkadzają bohaterowi nie tylko przeróżne ruchome ustrojstwa (tupiące buty w pierwszym poziomie, kapiąca woda w łazience, wyskakujące tost, pajaki i wiele innych), ale również wiele elementów tła (np. korona w pierwszym pokoju). To, co najbardziej ujmuję w tej grze, to fakt, iż każdy pokój żyje tu swoim życiem, posiada charakterystyczny zespół przeszkód i towarzyszące mu efekty dźwiękowe. Często ciekawie jest też rozwiązany problem ukrytego klucza. Na przykład w łazience zabranie korka implikuje spuszczenie wody z wanny, na dnie której leży właśnie ów cenny klucz. Grafika jest może i prosta, ale bardzo kolorowa i urozmaicona. A sama gra - choć momentami nieznośnie trudna - bardzo wciągająca. Ciekawa jest też geneza tytułu. Pierwotnie bowiem produkcja ta miała zostać wydana pod nazwą „**Home Sweet Home**”, jednakże ludzie z English Software chcieli wykorzystać fakt narodzin drugiego syna Księcia Karola i Księżnej Diany, Henryka (znanego obecnie jako Książę Harry). Podobno gra została (wraz z modelem C-64) wysłana do Buckingham Palace. Niespecjalnie wyobrażam sobie Księcia Karola grającego w jakąkolwiek grę komputerową, a Wy?

Na zakończenie tej części artykułu gra, która w zasadzie jedynie wykorzystuje pewne elementy jednoekranowych platformówek. Gdyż w **Space Taxi** wcielamy się w rolę - a to ci niespodzianka - kosmicznego taksówkarza. Autor programu, John Kutcher, dobrze odrobił lekcję z gier typu „**Lander**” (sam przyznaje się do inspiracji tytułem „**Lunar Lander**”) - sterowanie kosmiczną taksówką przypomina nieco manewrowanie pojazdami z tamtych starych produkcji. Z kolei z gier a la „**Manic Miner**” pożytych jednolity design poszczególnych poziomów, z nadaniem każdemu z nich unikalnej nazwy włącznie. Natomiast sama rozgrywka wygląda już zupełnie inaczej, niż w typowych jednoekranówkach. Na każdym z poziomów musimy bowiem zabrać na ogół kilku ludzików i wysadzić ich w miejscu, w którym sobie zażyczą. Aby ułatwić komunikację kierowca-pasażer, na wszystkich planszach zostało wydzielonych kilka platform, ponumerowanych od 1 do 9. Po przewiezieniu odpowiedniej liczby klientów przechodzimy do kolejnego poziomu.



O ile pierwszych kilka ekranów przejść jest dosyć łatwo, o tyle gra szybko staje się bardzo trudna - mamy coraz mniej miejsca na manewrowanie taksówką i pojawiają się różne ruchome elementy (np. w poziomie Pong Taxi jest to piłeczka pingpongowa). Bywa, że i same platformy się poruszają, a także że coś zakłóca sterowaniem pojazdu. Oprócz uważnego lotu i ostrożnego lądowania, należy pilnować poziomu paliwa no i uważać, by nie zahaczyć o potencjalnego klienta. Autor w projektowaniu poziomów wykazał się sporą wyobraźnią, gra potrafi więc wciągnąć. Dodatkowo po pokonaniu wszystkich 24 plansz przechodzimy do etapu pt. „mystery screen”, gdzie istnieje możliwość uzyskania dostępu do tajemnego menu. Gra zebrała plon bardzo dobrych recenzji, była również nominowana w kategorii „gra roku” pewnego magazynu. Nic dziwnego zatem, że „**Space Taxi**” cieszył i wciąż chyba cieszy się sporym powodzeniem. W 2004 roku został wydany sequel, „**Space Taxi 2**”. Aby wzbudzić większe zainteresowanie tą produkcją, wydano również „**Space Taxi Remake**”, zawierający pierwszych 8 poziomów oryginalnej produkcji Kutchera. Gra również w oczywisty sposób zainspirowała twórców tytułu z 1992 roku, o onomatopiecznym tytule „**Ugh!**”.

Z kosmicznej taksówki dość łatwo przejść się do bolidów F1. Wydaje mi się, iż wystarczy przywołać dwa tytuły „wyścigówek”, by dobrze zobrazować zjawiska, jakimi rządził się biznes gier komputerowych w 1984 roku. Gdyż z jednej strony mamy dopieszczoną graficznie, świeżą produkcję ekipy z Epyx, **Pitstop II**, z drugiej - **Pole Position**, nieco chyba spóźnioną konwersję klasycznej gry z automatu.



Spóźnioną chociażby dlatego, iż oryginalnie gra została wydana przez firmę Namco w 1982 roku. A to, co było świeże i oryginalne w roku 1982 (gra spopularyzowała - nie będąc jednak w tej kwestii pionierem - widok od tyłu na prowadzony samochód a także pseudo-trójwymiarową grafikę), dwa lata później już nie zaskakiwało. Chyba, że negatywnie - grafika nie mogła powalać nawet w 1984. I w tym miejscu miałem napisać, iż z grywalnością również nie jest najlepiej. Bo przecież możliwości oferowane przez „**Pole Position**” są niezwykle skąpe (wybieramy tylko typ Grand Prix i liczbę okrążeń), a sama jazda przedstawia się dość topornie i mało realistycznie. Niemniej gdy powróciłem do tego tytułu po wielu latach, stwierdziłem, że wciąż ma ona w sobie to magiczne coś, co nie pozwala zbyt szybko oderwać się od monitora.

Kilka słów o rozgrywce. W „**Pole Position**” musimy przede wszystkim pilnować, by w zadanym czasie pokonać każde okrążenie. Tym samym musimy wystrzegać się kraks, które powodują stratę kilku cennych sekund (nie ma natomiast ograniczeń dotyczących liczby kraks). Plusem gry są kwalifikacje, w których czas pokonania jednego okrążenia decyduje o pozycji startowej we właściwym wyścigu. W 1988 wydano sequel, „**Pole Position II**” (na automatach pojawił się już w 1983), z lepszą grafiką i większą ilością opcji. Oryginalny produkt Namco przewiwał się natomiast przez wiele różnych platform, od innych ośmiobitowców poczynając, na iPodach i współczesnych konsolach do gier kończąc (tu tytuł pojawił się w kolekcji pt. „**Namco Museum**”).

Jednakże o ile „**Pole Position**” mogła rywalizować - z powodzeniem - z wydaną w 1983 roku „**Pitstop**”, o tyle gra firmy Namco w pojedynku z sequelem nie miała już raczej szans. Powodów jest kilka. Po pierwsze - dru-

ga część „Pitstop” jest (jak na 1984 rok) bardzo dopieszczona graficznie. Po drugie – samo sterowanie pojazdem jest tu bardziej realistyczne – w „Pole Position” siła odśrodkowa była co najmniej za słaba. Jednakże tym, co zapewniło grze nieśmiertelność, jest podział pola gry na dwie części tak, by każdy z graczy widział aktualnie pokonywany fragment trasy.



Co najważniejsze, dzięki temu możliwa jest bezpośrednia rywalizacja z drugim graczem/komputerem – można np. zjechać przeciwnikowi drogę czy też „przepychać” się z nim na torze. Poza tym zachowano podstawowe reguły, znane z pierwszej części gry. Nie ma więc w „Pit Stop II” kraks, chociaż każdy kontakt z innym pojazdem w istotny sposób wpływa na zużycie opon oraz prędkość bolidu. Musimy też obserwować wskaźnik paliwa, zaś wszelkie wady pojazdu możemy skorygować po każdym okrążeniu w pit stopie. I w sumie łatwo zrozumieć, dlaczego produkcja firmy Epyx cieszy się dziś zasłużonym statusem jednej z najlepszych wyścigówek, jaka kiedykolwiek powstała na C-64.

Pozostańmy przy tematyce sportowej. Dla tego typu gier rok 1984 roku był znakomity. I tak na przykład koszykówkę reprezentowały takie tytuły jak „International Basketball” czy „One on One”. Ukazały się dwie niezłe gry z tenisem w roli głównej: „On Court Tennis” i „Match Point” (zresztą, kto wie, czy ten ostatni tytuł to nie najlepszy symulator tenisa na C-64). W tym miejscu chciałbym jednak przypomnieć zjawisko, które w 1984 roku było czymś nowym – gry sportowe typu multi events.

Zacznijmy od jednej z najsłynniejszych, a jednocześnie najbardziej prymitywnych gier sportowych wszech czasów. Chociaż nazwa **Decathlon** oznacza po prostu dziesięciobój, stała się synonimem energicznego machania joystickiem w lewo-prawo. Bo w taki właśnie sposób w grach tego typu zawodnik zyskiwał większą prędkość. Jakie dyscypliny wchodziły w skład dziesięcioboju? Z biegów mamy 100m płaskie i 110 przez płotki, następnie dystans średni – 400m i kończący rywalizację, wyczerpujący bieg na 1500 m. Dla lubiących skoki „Decathlon” oferuje skok w dal, wzwyż i o tyczce. Dziesiątkę uzupełniają konkurencje rzutowe – pchnięcie kulą, rzut dyskiem i oszczepem.



Pomijając prymitywizm sterowania, których jednych odrzuca, innych – zachwycia, grze brakuje nieco atmosfery. Duża w tym wina zbyt oszczędnej grafiki (nie narysowano np. piasku w skoku w dal). Wydana przez Activision gra ma jednakże ogromną zaletę – mieści się cała w jednym pliku, więc swego czasu mogli w nią zagrać także posiadacze wyłącznie magnetofonów. Zamiast stacji dysków graczom przydały się natomiast zapasowe joysticki i wytrzymałe mięśnie rąk – gdyż rozgrywka w „Decathlon” jest wyczerpująca dla obu.

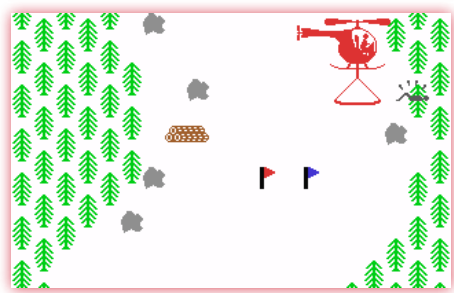
Największym wydarzeniem było jednak wydanie przez niezmordowaną ekipę z Epyx pierwszej części swojej serii olimpijskiej – **Summer Games**. Wystarczy spojrzeć na listę osób zaangażowanych w ten projekt, by zgodzić się, że produkcja gier powoli przestawała być biznesem chałupniczym. Gdyż pod „Summer Games” podpisało się aż siedmiu twórców. Jeden z użytkowników lemon64.com nazwał tę grę prawdziwym klasykiem, gdyż „rozpoczęła ona wszystko” – kładzenie nacisku na ładną grafikę, namiętne korzystanie z możliwości doładowania z dyskietki określonego fragmentu gry i... zalew gier „olimpijskich”, których od czasu „Summer Games” powstało faktycznie mnóstwo.



Sam Epyx wydał jeszcze m.in. „Winter Games”, „World Games” czy „California Games”. Inne firmy rozgrywki olimpijskie umieszczały np. w epoce jaskiniowców (Cave-man Ugh-Lympics), w cyrku (Circus Games) czy też w czasach rycerskich (Knight Games). Wymieniać można by długo. Natomiast wracając do „Summer Games”... Jak wskazuje nazwa, gra umożliwia wcielenie się w rolę sportowca uczestniczącego w letniej olimpiadzie. Dyscyplin – tak jak i w późniejszych produkcjach Epyx – jest osiem. Dwie konkurencje biegowe (100 m i 4x400m) oraz skok o tyczce wyczerpują temat „lekka atletyka”. Poza

tym dwukrotnie odwiedzamy basen (ponownie – 100m i sztafeta). Pozostałe konkurencje to skok przez kozła, strzelanie do rzutków i skoki do wody. I tylko w biegu na 100 metrów mamy rozwiązanie a la „Decathlon”. W pozostałych autorzy gry postawili na określone ruchy joyem w określonym czasie, więc tym razem obejdzie się bez treningu bicepsów. No i jednocześnie na rozgrywanie gry potrzeba jednak nieco więcej czasu (tu polecam zajrzeć do instrukcji). Jakie wrażenia wywołuje „Summer Games” dziś? Grafika w późniejszych olimpiadach Epyxu była lepsza, lecz i tu może się podobać – jest bardzo kolorowa. Gorzej z dźwiękiem, chociaż plusem tych gier jest możliwość wysłuchania wielu hymnów narodowych. A grywalność? Cóż, osobiście opowiadam się za teorią, iż większość gier sportowych z C-64 dosyć mocno trąci dziś myszką i „Summer Games”... nie jest pod tym względem wyjątkiem. Tak to przynajmniej wygląda w przypadku samodzielnej rozgrywki. Tym bardziej, iż nie wiedzieć czemu komputerowy zawodnik nie towarzyszy nam na basenie. Dziwne rzeczy dzieją się też na bieżni, gdy zawodnik drepta w miejscu. Natomiast z drugim graczem (a łącznie „Summer Games” umożliwia grę aż ośmiu zawodnikom) tego typu produkcje już chyba na zawsze pozostaną źródłem sporej zabawy. I tu również „Summer Games” wyjątkiem nie jest. Należy jeszcze dodać, iż nie jest to jedyna Olimpijska gra, która ujrzała światło dzienne w 1984 roku. Warto pamiętać również o „HES Games”. Gra wyróżnia się ciekawą grafiką (naprawdę gigantyczne postacie zawodników), nieco zabawną animacją, a także chyba ciekawszym doбором konkurencji – polecam zwłaszcza strzelanie z łuku. Pionierskim pomysłem było tu natomiast wykorzystanie powtórek a także możliwość zobaczenia rekordów świata. Chociaż gra jest dzisiaj nieco zapomniana, prawdopodobnie zainspirowała ludzi z Epyxu, którzy w swoich późniejszych olimpiadach umieścili – obecne w „HES Games” – takie dyscypliny jak podnoszenie ciężarów czy wspomniane już łucznictwo.

A co ze sportami zimowymi? Cóż, Epyx swoje „Winter Games” wydał dopiero w 1985. Póki co wszystkim graczom musiała wystarczyć prościuteńka gra, charakteryzująca się jednak całkiem wysoką grywalnością.



Tytuł wydany przez Mr. Chip Software, **Olympic Skier**, pozwala na wcielenie się w tytułowego „olimpijskiego narciarza”, uczestniczącego w trzech różnych konkurencjach: slalomie, skokach i zjeździe. Za każdą konkurencję dostajemy liczbę punktów uzależnioną od czasu pokonania trasy bądź uzyskanych metrów. W sumie w całej grze można uzyskać aż 1000

takich punktów, jednakże zdobycie chociażby połowy z nich nie należy do prostych zadań. Powód? Niezwykle trudno jest poprawnie pokonać całą trasę slalomu czy zjazdu na dużej prędkości. Dodatkowo, w przypadku zjazdu, nie dość, że trzeba omijać różne przeszkody, to jeszcze niektóre przesmyki między drzewami są ślepo zakończone. Pomimo bardzo prostej

grafiki i muzyki, „**Olympic Skier**” naprawdę wciąga – pokusa uzyskania coraz to lepszego rezultatu jest po prostu zbyt silna.

Na dziś to tyle. O tym, co też w 1984 roku działo się ciekawego w strzelankach bądź komnatówkach postaram się opowiedzieć w kolejnym odcinku.

p.a.

Booga-Boo

(The Flea)

Przygotowując się do napisania artykułu o 1983 roku, przypominałem sobie o grze, która niegdyś - pod nieco błędnym tytułem - gościła na jednej z moich kaset. **Booga-Boo (The Flea)** - bo o tej grze mowa - to bardzo stara, lecz bardzo sprytnie skonstruowana platformówka, w której wykorzystano scrolling, co w 1983 roku było rzadko spotykaną nowinką. Dla mnie jest też ona przykładem swoistego geniuszu twórców wczesnych gier, którzy z bardzo prostego pomysłu, z wykorzystaniem bardzo skąpych możliwości sterowania, zdołali wycisnąć tak wiele.

Gra wyróżnia się również swym bohaterem. Wcielamy się tu bowiem w rolę... pchły, która wpadła w pewną głęboką czelusć.

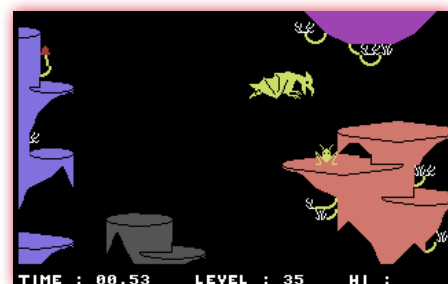


Tym samym cel gry jest bardzo prosty - musimy pomóc **Booga-Boo** wydostać się z tarapatów. Nie mamy na to zbyt wiele czasu - podziemia są bowiem zamieszkiwane przez smoka, który prędzej czy później wytropi pchłę. Aby wydostać się na samą górę, bohaterka musi wykorzystać znajdujące się tu i ówdzie platformy, krawędzie i tym podobne. Sterowanie jest wybitnie minimalistyczne. **Booga-Boo**

może po prostu skakać w prawo bądź lewo. W zależności od tego, jak długo przytrzymamy joystick/klawisz kursora, skok będzie krótszy bądź dłuższy. Siłę skoku reprezentują pokazujące się na dole ekranu kropki, w praktyce jednak należy kierować się intuicją, bo na kontrolowanie kropek zwyczajnie nie ma czasu. Na dole ekranu można również znaleźć informację, na jakiej wysokości się aktualnie znajdujemy (level), a także ile czasu zdążyło już upłynąć. Co jeszcze? Wykorzystanie kombinacji „fire+kierunek” pozwala na przesunięcie ekranu w danym kierunku, co ma niebagatelne znaczenie w obraniu właściwej drogi.



Pole gry nie jest zbyt obszerne, droga do góry - zbyt daleka. Nie oznacza to jednak, że prędko uporamy się z zadaniem uwolnienia pchły. Gdyż niełatwo jest dobrać odpowiednią siłę skoku tak, by wylądować akurat w tym miejscu, w którym chcemy. Bardzo łatwo natomiast po jednym nieudanym skoku spaść ze sporej już wysokości na sam dół, co zmusza pchłę do podjęcia wysiłku na nowo. Na szczęście upadki dla bohaterki gry nie są groźne. Nietrudno też ześlizgnąć się z jednej z platform prosto w mięsożerną roślinkę, bądź też natknąć się na smoka.



Tym samym gra może zirytować. Ale też wciągnąć na dłużej, niż moglibyśmy na samym początku podejrzewać. Oczywiście to rozrywka na jeden raz, do pierwszego ukończenia gry. Ale ten jeden raz zdecydowanie warto się z tym tytułem zmierzyć. Generalnie pozytywne wrażenia wzmacnia też bardzo ładna - jak na 1983 rok - grafika. Wszystko ładnie narysowane, wyraźne, czytelne. Gorzej z dźwiękiem, który lepiej wyłączyć. Polecam jednak „**Booga-Boo (The Flea)**” wszystkim miłośnikom tych najstarszych produkcji z C-64 i tym, którym wydaje się, że platformówki z tamtych lat to wyłącznie wariacje na temat „**Manic Miner**’a” bądź „**Donkey Kong**’a”.

p.a.

Producent i wydawca: Quicksilva

Rok: 1983

OCENA

Grafika: 55

Dźwięk: 25

Grywalność: 50

Ogółem: 50

Tales of the Arabian Nights

Podaję, że nie jestem jedyną osobą, której Ian Gray kojarzy się z wieloma przyjemnymi chwilami spędzonymi przed ekranem monitora. To właśnie ten programista podpisał się pod większością gier z serii „Dizzy”, a także kilkoma innymi zręcznościowymi przygodówkami, wydanymi przez niezastąpioną w takich przypadkach Codemasters. Jednakże przygoda Graya z C-64 rozpoczęła się już w okolicach 1983 roku, kiedy to dla Interceptor Software stworzył wiele gier, najczęściej dosyć prostych zręcznościówek. Wydana w 1984 roku „Tales of the Arabian Nights” to chyba najbardziej udana pozycja z tego okresu działalności Graya w branży gier komputerowych.



Fabula została zainspirowana Baśnią Tysiąca i Jednej Nocy. Gracz wciela się w rolę księcia Imrahila, który spieszy na ratunek swojej siostrze, porwanej przez złego sułtana Saladina. Gra składa się z ośmiu nocy - poziomów, które można z grubsza podzielić na dwie kategorie. Noce nr 1,3,5,6,7 to nie do końca typowe jednoekranowe platformówki. Zadanie jest zatem pozornie proste - trzeba zebrać wszystkie przedmioty znajdujące się na planszy, unikając różnorodnych przeszkód, a także upadków ze zbyt dużej wysokości. W tym przypadku kolekcjonujemy garnce z naniesionymi nań literami. Oryginalność „Tales of the Arabian Nights” polega na tym, iż musimy je zbierać tak, by litery utworzyły wyraz „ARABIAN”. Pozostałe poziomy to w zasadzie horyzontalnie zorientowana strzelanka, której akcja toczy

się na rzece (noc nr 2) bądź w powietrzu, podczas lotu magicznym dywanem (pozostałe poziomy).

To bardzo trudna gra. Szczególnie dają się we znaki poziomy z kolekcjonowaniem liter. W osiągnięciu celu przeszkadzają nam m.in. nieznosne ptaki, strzelające armatki, strzały z łuku, spadające kamienie czy też dzidy strażników. Jednym z ciekawszych poziomów jest ten oznaczony cyfrą 3, w którym musimy zmierzyć się z dzinnami pewnego czarnoksiężnika.



Nie trzeba też chyba nadmieniać, iż wymuszenie na gracz, by zbierał garnki w określonej kolejności, mocno komplikuje postawione przed nim zadanie. Pójście na przebój w przypadku tych poziomów oznacza z reguły szybką śmierć bohatera. A chociaż pomylić można się aż czterokrotnie, rychło można dojść do wniosku, iż życie jest i tak za mało.

Lepszą metodą na ukończenie tych poziomów jest znalezienie względnie bezpiecznego miejsca w obrębie danej planszy, by chociaż przez chwilę poobserwować, w jakim rytmie uaktywniają się poszczególne przeszkadzajki. W porównaniu z tymi poziomami, etapy „strzelankowe” przypominają wręcz rundy bonusowe. Przeprowadzenie przez rzekę jest tak proste, że nie wymaga specjalnego omówienia. Z kolei poziomy 4 i 8 łatwo przejść, przenosząc latający dywan w okolice prawego górnego rogu ekranu.



Grafika jest kolorowa i w gruncie rzeczy przyzwoita - trudno wymagać więcej od tak wiekowej pozycji. Nieco gorzej z muzyką, która przy tak trudnej rozgrywce może miejscami męczyc. Wydaje mi się, iż w tym konkretnym przypadku wykorzystanie kompozycji Rimskiego-Korsakowa nie było najlepszym pomysłem. Ćwierć wieku temu pewnym magnesem przyciągających graczy mogły być wstawki digitalizowanej mowy, w których narrator snuje opowieść Imrahila i oprowadza nas po poszczególnych poziomach. Oczywiście trudno, by wstawki te imponowały dziś w tym samym stopniu, co niegdyś, niemniej sam pomysł na takie przerwinki bardzo mi się podoba. Pomimo znacznego stopnia trudności oraz faktu, że podobnych gier trochę jednak powstało, jest w „Tales of the Arabian Nights” coś, co przyciąga na nieco dłużej. Może to wyraźnie wyczuwalny arabski posmak całości? Może pewne zróżnicowanie poziomów? Niezależnie od odpowiedzi, naprawdę warto poświęcić dłuższą chwilę na tą grę. Nawet jeśli po kilku godzinach rzucicie zirykowani joystickiem o ścianę.

p.a.

Producent i wydawca: Interceptor Software

Rok: 1984

OCENA

Grafika: 55

Dźwięk: 50

Grywalność: 55

Ogółem: 55

COMMODORE & AMIGA

widziane i zasłyszane

16 Bit - Changing Minds

Teledysk niemieckiej grupy **16 Bit** z wydanego w 1987 roku albumu **Inaxycvgtgb**

Użyto: Amiga 500 z monitorem 1081

Link: <http://www.youtube.com/watch?v=9aU3VXoRLVs>



23 - Nichts ist so wie es scheint

Film o dwóch hackerach, którzy za pomocą komputerów Commodore włamują się do systemów komputerowych. Za pośrednictwem przestępców tajne dane trafiają w ręce KGB.

Użyto: VIC-20, CBM-II, Commodore C64 I SX-64. Monitor 1802. Stacje dysków 1541.

Link: <http://www.youtube.com/watch?v=0rI2anb5Da0> (4:25 i dalej)



Airplane 2 / Czy leci z nami pilot 2

Druga część zwariowanej komedii. Komputery Commodore używane do kontroli lotu.

Użyto: Commodore C64.

Link: <http://www.youtube.com/watch?v=CTIkdBXe8g0> (1:06)



ALF

Popularny serial komediowy. Możemy w nim zobaczyć SX-64 w samolocie prezydenta USA oraz Amigę 2000

Użyto: Commodore SX-64, Amiga 2000

Link: <http://www.youtube.com/watch?v=ezx4exkbIZ8> (1:13)



Apoptygma Berzerk - Kathy's Song

Norwescy muzycy z zespołu Apoptygma Berzerk grający alternatywnego rocka na okładce swego singla z 2000 roku wykożystali Kathy's Song (Come Lie Next To Me) wykożystali Commodore 64 wraz z osprzetem, natomiast w utworze możemy również usłyszeć nawiązania do dźwięków SID'a.

Użyto: Commodore C64, magnetofon 1530, joystick Competition Pro i monitor 1802.

Link: http://www.youtube.com/watch?v=Ci_LIavsEhQ



Informacje znalezione w Internecie. Uporządkował, przetłumaczył i wyszukał na YT - Black Light. CDN

Mimic Systems' Spartan

Szukając po Internecie tematu na nowy artykuł, natrafiłem na krótki opis sprzętowego emulatora komputera Apple II. Zaintrygował mnie fragment o tym urządzeniu, więc postanowiłem dowiedzieć się coś więcej i przy okazji napisać artykuł. Oto jak Brent Marykuca, główny programista Mimic Systems opowiada historię projektu Spartan'a:

W 1983 lub 84 roku zaraz po premierze C64 mój przyjaciel mieszkający w mieście Victoria na terenie Kolumbii Brytyjskiej (jest to prowincja Kanady) przerobił oryginalną grę na Apple II pod tytułem Space Invaders i dzięki temu mogła być uruchomiona i działała całkiem przyzwoicie na podobnie wyposażonym C64.



Fot. Space Invaders na komputerze Apple II

Komputer Apple II jest 8-bitowym komputerem domowym opracowanym już w latach 70-tych XX wieku. Komputer ten posiada

popularny w tamtych czasach procesor MOS 6502, który jest w dużej mierze kompatybilny z procesorem C64 6510. Posiada od 4KB do maksymalnie 48KB pamięci RAM. Rozdzielczość ekranu 280 x 192, 40 x 24 kolumn tekstu, tylko 6 kolorów. System operacyjny Woz Integer BASIC zawarty jest w ROM-ie. W tamtych czasach był bardzo popularnym komputerem i doczekał się kilku rozbudowanych wersji. Powstało też kilka klonów tego komputera. Dawny ZSRR zbudował komputer Agat, podobnie jak Bułgaria Pravec, które zostały oparte na rozwiązaniach zastosowanych właśnie w komputerze Apple II.



Fot. Komputer Apple II

W Australii powstał komputer kompatybilny z Apple II o nazwie Medfly, który miał znacznie szybszy procesor, więcej pamięci i kilka dodatkowych usprawnień. Jego nazwa odnosiła się do Apple, bo Medfly to skrót od nazwy Mediterranean fruit fly, jest to odmiana pewnej muchówki, która najczęściej atakuje

jabłka czyli po angielsku Apple.

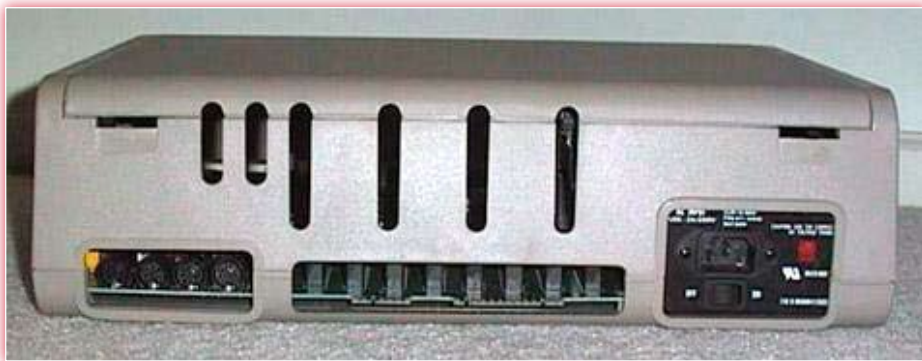
Wracając do dalszej opowieści: Naturalnie, perspektywa pisania programów które mogły być uruchamiane przez posiadaczy stosunkowo niedrogich C64, a pisanych pod Apple II była niezwykle obiecująca. Ludziom, których wtedy znałem, bardzo ten pomysł się spodobał. Wkrótce okazało się że nie można tego zrealizować jedynie drogą programową. Dlatego powstało rozwiązanie sprzętowe zrealizowane przez firmę Mimic Systems. W roku 1985 (miałem wtedy 19 lat) zostałem zatrudniony jako główny programista produktu nazwanego „The Spartan” – dublera Apple II. The Spartan był dokładnym klonem Apple II z jedną różnicą. Różnica polegała na braku klawiatury z Mac'a.



Fot. Spartan

Z przodu urządzenia znajdowała się seria złącz i kabli, które dokładnie pasowały do złącz na tylnej ścianie C64. Pomysł polegał na tym





Fot. Tył Spartan'a. Porty C64 były przejściowe i dlatego Spartan mógł być cały czas podłączony.

że The Spartan był niejako doklejany do C64 (lub c64 był dosłownie wydłużany o emulator). Wszystkie złącza były także na tylnej ścianie urządzenia, tak że użytkownik nie musiał z niczego rezygnować. W tym momencie dysponował 2 procesorowym systemem zdolnym uruchamiać wszystkie programy napisane na C64 i Apple II każdy z nich równolegle. Dodatkowy układ sterował przełączaniem klawiatury oraz wyświetlaniem obrazu pomiędzy trybami C64 i Apple II. Dodatkowo The Spartan udostępnił 4 sloty cartridge'a, podczas gdy w oryginalnym C64 był tylko jeden, jak również konwerter joysticka z C64 dla Apple II. Najgorszym ze wszystkich dodatków był tak zwany „DOS Card” czyli kontroler dyskiety który należało zainstalować wewnątrz Commodorowskiego napędu, pomiędzy mechanizmem stacji a jej płytą główną. W trybie stacji 1541 DOS-Card po prostu przepuszczał przez siebie wszystkie sygnały, ale po użyciu przełącznika przejmował kontrolę nad mechanizmem i stacja stawała się w pełni kompatybilnym napędem z Apple II. Jak łatwo sobie wyobrazić powodowało to liczne uszkodzenia dyskiety zarówno z Apple II jak i C64.

Spartan mógł być łatwo dostosowany do potrzeb użytkowników. Zawierał co najmniej 24 przełączniki na płycie głównej i karcie procesora. Użytkownicy mogli konfigurować takie elementy jak: przerwania, mapowanie ROMu, który komputer ma się zgłaszać zaraz po włączeniu zasilania, możliwości resetu (czy Spartan może resetować c64 i odwrotnie), mapowanie adresów pewnych szczególnych funkcji urządzenia, sposoby konwersji joysticka oraz dostępność banków dodatkowej pamięci RAM.

Do urządzenia dołączane było oprogramowanie, które zawierało Applesoft BASIC, który był w 100% kompatybilny ponieważ powstał przez zdzasemblowanie oryginalnego pliku binarnego z ROMu Apple II. Następnie pozmieniano kolejność poszczególnych rozkazów odpowiedzialnych za wyświetlanie obrazu, w celu spowodowania różnic względem oryginału. Dzięki temu zabiegowi ominięto prawa autorskie i niepotrzebne było wykupowanie licencji. Spartan posiadał programową kontrolę przełączania pomiędzy dźwiękiem i obrazem z obu kompu-

terów jak również tryb mixed-audio. Assembler i monitor dla komputerów Apple II dostępny był po stronie C64. Oprogramowanie zawierało tryb slave (jeden computer stawał się podrzędny i dokonywał obliczeń dla drugiego - głównego) aktywowany przez odpowiednie komendy z poziomu monitora obu maszyn. Polecenia te dostępne były także z poziomu BASIC'a obu maszyn. Można było przesyłać dane pomiędzy dwoma systemami, a nawet uruchamiać programy na komputerze przełączonym w tryb slave. Najciekawszym przykładem użycia tej funkcji było demo z efektami 3D w którym niektóre obliczenia były wykonywane przez komputer w trybie slave.

Spartan był sprzedawany na przełomie lat 1985/86. Oprócz pewnych wyzwań technologicznych cała historia Mimic Systems była wstrętą. Charakterystyczne były cotygodniowe zmiany designu urządzenia przez głównego szefa firmy. Oczywiście bywało również że po tygodniu wycofywał wszystkie poczynione zmiany. W firmie pracował kompletnie zdemotywowany, wypalony zespół inżynierów. Pamiętam jak zatrudniono trzy osoby jednego dnia. Jednego gościa tylko dlatego, że obserwował plotter rysujący schemat jakiegoś układu scalonego. Dodatkowo dawał się we znaki drański system zarządzania (zespół był płacony od przepracowanych godzin, ale jednocześnie odbijane były wszystkie wyjścia do toalet). Dla mnie historia urządzenia zakończyła się na początku 1986 roku. Odszedłem szybko kiedy szef wypisał sobie wielki czek z konta z którego szły wypłaty i uciekł – według plotki do południowej Ameryki.



Fot. Płyta główna Spartan'a

Wszystkie złącza i wtyczki DIN z przodu pasowały do odpowiednich złączy na tylnej ścianie C64. Z prawej strony obudowy znajduje się zewnętrzne złącze cartridge'a C64 (3 pozostałe znajdują się w środku obudowy. Widać również 3 przyciski reset (C64, C64 z wybranym cartridge'm i Spartan) Te 3 przyciski reset w instrukcji obsługi opisane były jako najważniejsza i prawdopodobnie najbardziej funkcjonalna część systemu Spartan.



Fot. Wnętrze pokrywy z widocznymi podpisanymi twórców.



Fot. Pierwsza reklama Spartan'a, ukazała się w magazynie "Ahoy!"

Pomysł był dobry, jednak koszt był przerażający. 599 dolarów na tamte czasy było bardzo dużą kwotą. Zbyt duża cena odstraszała potencjalnych klientów i tak naprawdę mało osób decydowało się na jego kupno, przez co urządzenie musiało zniknąć z rynku i zostało całkowicie zapomniane. Obecnie jest rzadkością wśród kolekcjonerów sprzętu spod znaku Commodore. Szkoda, że cena projektu była zaporą nie do przekroczenia dla większości potencjalnych klientów. Spowodowało to szybkie zniknięcie tego urządzenia z rynku. Gdyby projekt nie został zniszczony ekonomicznością, na pewno wkrótce można by było w ten sam sposób emulować inne komputery np. Spectrum czy Atari.

MrMat & Ramos

+60k dla C128

Mr. Fiz/Samar zaprojektował rozszerzenie pamięci dla C64, o nazwie +60k. Stało się ono de facto standardem na polskiej scenie. Opiszę, jak w prosty sposób zmodyfikować C128 tak, aby w trybie C64 emulowane było zachowanie rozszerzenia +60k z wykorzystaniem już fabrycznie zamontowanych kości pamięci.

Zasada działania

W wersji dla C64 instalacja rozszerzenia polega na zamontowaniu dwóch dodatkowych kości pamięci oraz układu zarządzającego dostępem do nich. Przestrzeń adresowa VICa (\$D000-\$D3FF) zostaje podzielona na 4 strony pamięci. Pierwsza strona (obszar \$D000-\$D0FF) nadal odpowiada za VIC, następna (\$D100-\$D1FF) aktywuje zatrask kontrolujący wybór banku pamięci +60k, a kolejne dwie (\$D200-\$D2FF oraz \$D300-\$D3FF) są niewykorzystane.

Rejestrem sterującym rozszerzeniem pamięci jest najstarszy bit komórki \$D100. Co ważne - jest to rejestr tylko do zapisu. Z tego powodu prawidłową metodą programowania +60k jest tylko zapis wartości \$00 lub \$80 do komórki \$D100, a nie używanie np. rozkazów INC/DEC \$D100 lub testowanie odczytanej wartości.

Rejestr kontrolny decyduje o tym, który bank pamięci będzie widziany przez procesor w obszarze \$1000-\$FFFF. To 60KB RAM, stąd nazwa układu. Przełączanie banków pamięci ma wpływ tylko na CPU. VIC zawsze pobiera dane z pierwszego banku pamięci (wartość \$00 w \$D100), ustawionego domyślnie po resecie komputera.

Wersja dla C128

C128 jest wyposażony w dosyć zaawansowany kontroler pamięci (MMU – Memory Management Unit), który pozwala na rozmaite konfiguracje dostępu do pamięci. W trybie C64 ten układ jest jednak całkowicie niedostępny dla programisty i z naszego punktu widzenia bezużyteczny. Bez modyfikacji sprzętowej dostęp do obu banków RAM w trybie C64 nie jest możliwy.

Przy konstruowaniu wersji dla C128 wziąć pod uwagę następujące czynniki:

- dodatkowa pamięć już znajduje się na płycie, w postaci dwóch lub ośmiu kości (zależnie od modelu C128)

- w trybie C128 rozszerzenie musi być nieaktywne – chcę pełnej zgodności z C128
- w trybie C64 rozszerzenie musi mieć możliwość dezaktywacji, aby móc uzyskać pełną zgodność z C64
- MMU nie wpływa na zarządzanie pamięcią w trybie C64

Układ

Części niezbędne do wykonania układu to:

- zatrask 74HCT174
- klucz 4066
- dekodery/demultipleksery 74HCT139
- bramki NAND 74HCT00
- bramki NOR 74HCT02
- inwerter 74HCT04
- przełącznik
- przewody (używałem pociętej i rozdzielonej taśmy wielożyłowej)

Proponuję użyć układów serii HCT. Zamiast 74'04 można wziąć 74'00 lub 74'02 i połączyć oba wejścia jednej z bramek tak, aby uzyskać funkcję NOT.

Układ składa się z dwóch bloków. Pierwszy blok odpowiada za podział przestrzeni adresowej VIC i rejestr sterujący (przełącznik, połowa 74'139, zatrask i dwie bramki NOR). Sposób działania jest następujący: jeżeli procesor operuje na przestrzeni adresowej VIC (\$D000-\$D3FF), to linia VIC/CS przechodzi w stan niski i aktywuje dekodery '139. Tutaj linia adresowa A8 (ósmą bit adresu) kontroluje, które z wyjść dekodera będzie w stanie aktywnym: to podłączone bezpośrednio do VICa (dla strony pamięci \$D0xx), czy też wejście zatrasku (dla strony \$D1xx). Jeżeli nastąpił zapis, co jest sygnalizowane przez linię R/W z procesora, to wartość z linii danych D7 zostanie zapamiętana na wyjściu zatrasku (pin 7 układu 74'174).

Drugi blok kontroluje dostęp do pamięci w obszarze \$1000-\$FFFF i składa się z pozostałych elementów: bramek NOR, NAND, drugiej połowy 74'139, klucza 4066 i inwertera). Jeżeli następuje operacja na obszarze \$1000-\$FFFF, czyli wszystkie linie adresowe A12-A15 są w stanie wysokim i jeżeli operacji tej dokonuje CPU (linia AEC), to na wyjściach drugiego dekodera '139 pojawi się stan odpowiadający wartości logicznej zapamiętanej na zatrasku. Linią aktywującą dekodery jest /RAMCAS0. Ten sygnał przechodzi w stan niski, gdy jest żądany dostęp do pamięci. Ponieważ jednak w

trybie C128 chcemy całkowitej kompatybilności z oryginalnym sprzętem, ostateczny sygnał dostępu do pamięci (/RAMCAS0 dla banku 0 lub /RAMCAS1 dla banku 1) jest przekazywany za pośrednictwem klucza 4066 i pochodzi albo z naszego układu dekodera, albo z MMU, w zależności od sygnału na linii /64/128. W trybie C128 klucz 4066 skutecznie odseparuje układ +60k od kontroli nad kośćmi RAM.

Schemat jest przedstawiony na rysunku. Uwaga – zasilanie zostało tutaj pominięte! Nie wolno zapomnieć o jego podłączeniu do wszystkich instalowanych kości.

Montaż

Niektóre nazwy sygnałów na schemacie zostały określone jako CHIP/BOARD. Oznacza to, że dany układ trzeba wyjąć z podstawki i wygiąć odpowiedni pin tak, aby nie miał kontaktu, gdy kość wróci na płytę główną. Sygnały CHIP należy pobrać z przewodów przylutowanych bezpośrednio do wygiętych pinów, a sygnały BOARD z odpowiadających im kontaktów na płycie głównej.

Tam, gdzie to możliwe podałem kilka źródeł pozyskania danego sygnału. W zależności od modelu C128 może to ułatwić montaż.

Z kolei dla sygnału VIC/CS jako BOARD oznaczyłem U11-42, czyli 42. nóżkę układu U11, a nie U21-19, czyli punkt lutowania 19. nóżki VIC. To dokładnie ta sama linia. W trakcie montażu okazało się, że łatwiej pobrać go z U11, który był po prostu bliżej. **Uwaga przy wyjmowaniu U7 (MMU) i U21 (VIC) z podstawek!** To duże kości i łatwo je uszkodzić lub wręcz pourywać nóżki w trakcie tej operacji. To nieco kłopotliwe zwłaszcza przy VICu, który znajduje się wewnątrz metalowej puszkii na płycie głównej komputera.

Polecam też szczególną ostrożność przy lutowaniu przewodów do układów scalonych. Trzeba to wykonać na tyle szybko, aby zbyt długo ich nie podgrzać, ani nie doprowadzić do zwarcia. Warto wcześniej potrenować, mieć odrysować do cyny w pogotowiu i nie zapomnieć o użyciu topnika.

Przełącznik przekazuje sygnał A8 lub GND do wejścia dekodera 74'139. Trzeba umieścić go gdzieś na obudowie komputera. Na pewno się przyda, ponieważ istnieje oprogramowanie, przy którym obecność +60k sprawia problemy.

Kości nowych układów scalonych najłatwiej zamontować odginając wszystkie nóżki z wyjątkiem GND i +5V w bok, a następnie przylutowując je na podobnych układach już znajdujących się na płycie C128. Np. dla bramek 74'00, '02 i '04 oznacza to odgięcie nóżek z wyjątkiem 7 (GND) i 14 (+5V). Pozwala to na bezpieczne i trwałe połączenie bez potrzeby wykonywania dodatkowej płytki.

Po zamontowaniu części odpowiedzialnej za rejestr sterujący można sprawdzić, czy

układ zachowuje się poprawnie w trybie C64. Jeżeli rozszerzenie jest wyłączone przełącznikiem, to w \$D100-\$D200 powinna być kopia rejestrów VICa (zapis do \$D120 zmieni kolor ramki). Jeżeli rozszerzenie jest aktywne, to powinniśmy móc zmieniać stan logiczny na pinie 7 zatrzasku 74'174 zapisem wartości \$00 lub \$80 pod adres \$D100. Do kontroli tych funkcji wystarczy kilka instrukcji POKE z poziomu BASICa. Pozostała do wykonania praca nad pozostałą częścią układu nie powinna już sprawiać problemów.

Podsumowanie

Projekt nie jest trudny i do jego wykonania nie są potrzebne żadne kłopotliwe w znalezieniu elementy. W efekcie nasz C128 zyskuje niewielkim kosztem nową funkcjonalność, a sami otrzymujemy dostęp do dem i narzędzi zgodnych z +60k. Dobrej zabawy i powodzenia!

Maciej 'YTM/Elysium' Witkowiak,
ytm@elysium.pl, I.2010

Dual Drive Burst

Dawno temu mój scenowy kolega, Szymon 'Zyga/Alliance' Zygański, zadał proste pytanie: jak można wykorzystać dwie stacje dysków, obie wyposażone w burst, do szybkiego kopiowania dysków? W ten sposób powstał Dual Drive Burst Backup, najszybszy znany mi kopier dysków.

Burst?

Jak wiemy, stacja 1541 nie jest demone. Dla swappera zwykłą operacją było kopiowanie całych dysków, które zwykle zajmowało sporo czasu. Dużym ułatwieniem było posiadanie drugiej stacji dyskietek – dzięki niej można było oszczędzić czas i energię na przekładanie dyskietek.

Powodem powolności 1541 są tragicznie napisane procedury szeregowej transmisji danych pomiędzy stacją dysków i komputerem. Problem został rozwiązany na co najmniej trzy sposoby:

- procedury fastloaderów, które przy perfekcyjnej synchronizacji 1541 i C64 przesyłają dane tak szybko, jak to tylko możliwe za pomocą standardowych kabli
- sprzętowy port szeregowy w C128 i stacjach 1571/1581, który osiąga podobną wydajność również za pomocą standardowych kabli
- zbudowanie łącza równoległego do przesyłania całych bajtów ze sprzętowym handshake (burst)

Istniały co prawda zamienniki ROMów Kernala i DOSu w stacji, które zmieniały ob-

sługę systemowej transmisji szeregowej na użycie kabla burst, jednak podstawowym zastosowaniem tej modyfikacji było błyskawiczne kopiowanie całych dysków.

Prędkość odczytu i zapisu kopierów burst jest niemal taka sama, jak przy szybkim formacie. Niestety co kilka sekund wymagają przekładania dyskietek. Transmisja jest tak szybka, ponieważ można przesyłać z/do stacji dysków surowe dane odczytywane bezpośrednio z głowicy. Dekodowanie GCR nie jest konieczne. Ograniczeniem jest natomiast ilość pamięci w C64. Gołe 64KB RAM to za mało, aby pomieścić cały obraz dyskietki. Rozszerzenia pamięci (RamCart lub +60k) pozwalały tylko zmniejszyć liczbę przełożeń dyskietek.

Dual Drive Burst

Pomysł Szymona był genialnie prosty. Należało po prostu połączyć ze sobą jedno-dwójnego wszystkie przewody kabli burst z dwóch stacji dysków i podpiąć je do User Portu. W zwykłej konfiguracji porty w stacjach i w C64 działają domyślnie jako wejścia, więc nie ma zagrożenia dla układów I/O.

Niedługo później powstał DDBB (Dual Drive Burst Backup). Program pozwala na kopiowanie burstem dyskietek bez potrzeby ich przekładania między stacjami. Śmiem twierdzić, że dzięki tej drobnej zmianie to najszybszy kopier dysków – licząc czas od uruchomienia kopiowania do jego ostatecznego zakończenia.

Program jest bardzo prosty w obsłudze. Najpierw należy wybrać urządzenie źródłowe i docelowe. Potem ukaże się nam ekran z mapą dysku. Klawiszami funkcyjnymi można wybrać opcję kopiowania (F1), skanowania dysku (F3), włączenia/wyłączenia weryfikacji zapisu (F5) i wyczyszczenia ekranu (F7).

Czy na tym koniec?

DDBB okazał się wystarczająco dobry dla mnie, jako swappera. Zabrakło więc już motywacji do stworzenia czegoś jeszcze szybszego.

Pomysł byłby następujący: niech C64 tylko inicjuje transmisję, a dane niech przepływają już bezpośrednio między obiema stacjami dysków – za pośrednictwem kabla burst z głowicy na głowicę. Czy to technicznie możliwe? Może wśród Czytelników znajdą się chętni na podjęcie wyzwania?

Źródła

Archiwum z DDBB oraz jego źródłami (na licencji GNU GPL) znajduje się na mojej stronie: <http://members.elysium.pl/ytm/html/projects.html> (u dołu).

Schemat układu burst (wg Mr. Fiz/Samar) również jest tam dostępny: <http://members.elysium.pl/ytm/gfx/workshop/projects/burst-scheme.png>

Maciej 'YTM/Elysium' Witkowiak,
ytm@elysium.pl, I.2010

Romuald Drahokaupil



Oto wspomnienia człowieka, który dla C64 w Polsce zrobił bardzo wiele. Niestety z powodu powszechnie panującego piractwa i nieprzestrzegania praw autorskich, których i tak nie było, nie odniósł zasłużonego sukcesu. Stworzył do C64 niezapomniane urządzenie, ułatwiające pracę z tym komputerem o nazwie „Black Box”, a także zaprojektował i wykonał kilka innych urządzeń, które spełniały przeznaczone dla nich zadania. Poniżej przedstawiamy wywiad z panem Romualdem Drahokaupilem i zapraszamy do przyjemnej lektury, przedstawiającej historię tego człowieka związaną z Commodore 64.

Dzień dobry panie Romualdzie, dziękuję, że zgodził się pan na udzielenie wywiadu i wspomnianie dawnych czasów. Większość osób nie ma pojęcia, że jest pan twórcą kartridża o nazwie Black Box, więc prosiłbym o krótkie opisanie swojej osoby.

Nazywam się Romuald Drahokaupil, mieszkam we Wrocławiu. Jestem absolwentem Wydziału Elektroniki Politechniki Wrocławskiej. Po ukończeniu studiów przez 22 lata pracowałem w tej uczelni, a konkretnie w Instytucie Telekomunikacji i Akustyki, na etacie naukowo-dydaktycznym. Tam też zrobiłem doktorat. Mimo, że obecnie nie pracuję, mam się dobrze i dysponuję dużą ilością czasu do realizacji własnych pasji. Pozostało mi już tylko niecałe trzy lata do emerytury, co w pośredni sposób wskazuje na mój wiek.

Większość osób myśli, że pańskie nazwisko kojarzy się z greckim nazwiskiem. Może pan wyjaśni skąd takie dziwne nazwisko u pana.

To nie jest nazwisko greckie ale czeskie;

„draho” to po czesku „drogo”, a „koupil” znaczy w tym języku „kupił”. Być może jakiś mój czeski praprzodek coś kiedyś drogo kupił i dlatego tak go nazwano. Jednak pisownia mego nazwiska stanowi pewną rzadką odmianę, ponieważ zamieniono w niej drugie „o” na „a”. Takie różne wersje tych samych nazwisk się zdarzają również w naszym kraju, np. Zamojski i Zamoy-ski. Nazwisko w wersji „Drahokaupil” obecnie noszą chyba tylko dwie osoby w Polsce – ja i moja żona. Niestety, moja znajomość własnej genealogii nie sięga tego Czecha, od którego się wszystko zaczęło. Wiem tylko tyle, że moja rodzina pochodzi, jak większość wrocławian, zza Buga, a te tereny, zwane niegdyś Galicją, przypadły Austrii podczas rozbiorów Polski. Za miłościwie panującego cesarza Franciszka Józefa do Austro-Węgier należała nie tylko Galicja, ale wiele innych krajów, w tym również i Czechy. Można było się po tych terenach swobodnie przemieszczać i prawdopodobnie jakiś Czech ożenił się z Polką, a następnie osiadł gdzieś na tych zabużańskich ziemiach. Musiało to jednak stać się bardzo dawno, bo już mój pradziadek uważał się za Polaka.

Skąd pomysł zainteresowania się elektroniką i połączenia go z komputerami?

Moje zainteresowanie elektroniką wynikało z mojego zawodu, ale fascynacja komputerami i programowaniem była czysto osobista i zupełnie prywatna, ponieważ zanim nabyłem C64, nie miałem najmniejszego pojęcia o komputerach; nigdy też przedtem nie napisałem żadnego, nawet najprostszego programu. Byłem zupełnym ignorantem w tych sprawach, ale miałem silne pragnienie zmiany stanu rzeczy. Commodore był moim pierwszym źródłem wiedzy na ten temat, której nie zdobywałem „pod przymusem”, jak to często zdarza się w szkole czy na studiach, ale którą chłonałem z własnego wyboru, przez co każdy postęp w tej dziedzinie dawał mi dużo satysfakcji. Szczególnie interesowało mnie połączenie hardware-software, ponieważ każde urządzenie współpracujące z komputerem musi być odpowiednio oprogramowane. Jako inżynier elektronik wiedziałem jak zrobić hardware i potrafiłem to wykonać, jednak nie znałem się na tym drugim i tego musiałem się właśnie nauczyć. Pierwszą rzeczą jaką skonstruowałem do C64 wcale nie był kartridż, a programator pamięci EPROM. Każdy kartridż ma wewnątrz EPROM, który trzeba zaprogramować przed wlutowaniem do płytki drukowanej, a ja nie miałem takiego pro-

gramatora. W drugiej połowie lat osiemdziesiątych ubiegłego wieku były co prawda już w Polsce dostępne programatory sterowane z PC, ale mnie nie było wówczas stać na kupno drogiego programatora i jeszcze o wiele droższego od niego peceta. Dlatego zbudowałem dość toporny, ale skuteczny programator sterowany z C64, co wzbogaciło ten komputer o jeszcze jedno zastosowanie. Urządzenie było przeznaczone do programowania pamięci do 32 KB i było połączone z C64 przez ten sam port co kartridże. Napisałem odpowiednie oprogramowanie, które po kilku korektach błędów przestało wreszcie psuć te pamięci i było gotowe do użycia. Teraz mogłem wreszcie zabrać się za kartridże.

Porozmawiajmy o czasach, kiedy w domach na biurkach zamiast pecetów, królował Commodore 64. Kiedy zainteresował się pan tym komputerem?

Chociaż pierwsze C64 zostały wypuszczone na rynek już w 1982 roku, w Polsce panował wtedy stan wojenny i wszyscy byli wówczas zaabsorbowani bardziej przyziemnymi, bytowymi sprawami, niż śledzeniem rozwoju komputeryzacji na Zachodzie. Jeśli już coś do nich stamtąd docierało, to raczej wiadomości Radia Wolna Europa, niż nowinki techniczne. Prasa, radio i telewizja zaczęły szerzej informować o komputerach osobistych i domowych z kilkuletnim opóźnieniem. Równocześnie pojedyncze osoby, które miały przywilej wyjazdu poza „żelazną kurtynę”, przywoziły ze sobą pojedyncze sztuki C64, ZX Spectrum lub Atari. Ja swój pierwszy C64 zamówiłem w zachodnioniemieckiej firmie handlowej, chyba w połowie lat osiemdziesiątych. Musiałem w tym celu założyć sobie konto walutowe i dokonać zagranicznego przelewu. Czekałem na dostawę kilka miesięcy, z czego większość czasu zajął właśnie ów przelew. Dlaczego wybrałem właśnie C64? Nie był to raczej świadomy i przemyślany wybór, oparłem się głównie na dobrych radach moich bardziej zorientowanych kolegów. Ponieważ wówczas na komputerach zupełnie się nie znałem, sprowadziłem C64 właśnie po to, aby się o nich czegoś dowiedzieć. Następnym etapem było zgromadzenie odpowiedniej literatury i oprogramowania. Największy kłopot sprawiały książki, bo na polskim rynku ich nie było i trzeba było grube tomy kserować i zszywać sznurkiem. Co do oprogramowania, to go nie brakowało, bo C64 egzystował na zachodnim rynku już kilka lat i trochę się tego nabrało.

Tworząc coś na C64 sam pan musiał dochodzić, gdzie co leży w pamięci czy miał pan już coś takiego jak mapa pamięci?

Mając program umożliwiający przeglądanie zawartości pamięci (takie programy nazywają się monitorami), można było niektóre rzeczy rozpoznać, np. wykryć obszar adresowy BASIC ROM i KERNAL ROM. Jednak do wielu rzeczy nie dawało się w ten sposób dojść, na przykład do informacji związanych z rejestrami generatora dźwięku (SID) czy obrazu (VIC). Do tego potrzebna była odpowiednia literatura. Na temat Commodore napisano chyba kilka książek, ale tylko jedna pozycja jest najważniejsza. Jest to: „Commodore 64 Programmer's Reference Guide”, Commodore Business Machines, Howard W. Sams & Co., Inc., 1982. Książkę tę nazywano „biblią Commodore” i właśnie z niej korzystałem. Można było tam znaleźć wszystko: mapy pamięci, opisy portów rozszerzeń i ich adresy, wykaz ważnych rejestrów, omówienie języka BASIC, wszystkie rozkazy języka maszynowego i wiele innych rzeczy, co pozwalało całkowicie „rozgryźć” ten komputer.

Jak większość posiadaczy C64 swoje kroki zaczynała od BASIC'a, czy pierwsze pana programy były w tym języku i co to było?

Oczywiście, to była naturalna kolej rzeczy dla kogoś, kto się zupełnie nie znał na komputerach. Napisałem wiele programów w tym języku, ale w większości były to programy szkoleniowe, poprzez które uczyłem się C64. Kilka z nich wykorzystałem nawet w swojej pracy naukowej, a kilka innych użyłem zdaje się przy oprogramowaniu kartridży, ale już nie pamiętam wszystkich szczegółów. Na pewno program demonstracyjny syntezy mowy wpisany do Black Box'a V.8, łącznie z wokalną aranżacją piosenki „Płynie Wisła płynie” oraz instrumentalnym wykonaniem utworu „Love me tender”, jest jednym z przykładów moich BASIC'owych dokonań. Jednak szybko stwierdziłem, że programy napisane w języku BASIC, zwłaszcza te bardziej złożone, są dość „mułowate”, ponieważ nie są kompilowane i działają na zasadzie interpretera, a to oznacza, że podczas wykonywania tego rodzaju programów są z nich pobierane kolejne dyrektywy i „na żywo” tłumaczone na ciąg rozkazów maszynowych, co bardzo spowalnia działanie takich programów. Dlatego po przejściowym okresie zainteresowania programowaniem w BASIC'u, w późniejszym okresie przeważającą większość czasu poświęciłem tworzeniu programów pisanych w języku asemblera.

Początki nauki asemblera na pewno były trudne? Jak wyglądała u pana jego nauka?

Te początki wcale nie były trudne, były ekscytujące. Działo się tak z dwóch powodów. Po pierwsze, nie uczyłem się asemblera bo musiałem, ale dlatego że chciałem. Po dru-

gie, dobrze się przy tym bawiłem. To nie była nauka według wytyczonego przez kogoś nudnego programu, ale samodzielne poznawanie działania komputera przez doświadczenie. W ten sposób znacznie łatwiej coś zrozumieć niż poprzez przyswajanie sobie cudzych myśli i gotowych algorytmów. Dla studenta, który ma tylko kilka dni do egzaminu i zarywa noce, by przyswoić sobie jak najwięcej wiadomości podręcznikowych, taka nauka może jawić się jako coś trudnego. Dla kogoś nieograniczonego obcymi wymaganiami, ani żadnym limitem czasowym, kto traktuje naukę jak dobrą rozrywkę, nie ma rzeczy trudnych. Zabawa jest najlepszym sposobem poznawania świata, o czym wie każde dziecko, które w ten sposób odkrywa praktyczne aspekty praw fizyki, uczy się mechaniki ruchu, poznaje właściwości różnych substancji, uczy się posługiwać własnym ciałem i poznaje w radosny sposób wiele innych rzeczy. Dzieci do zabawy zmuszać nie trzeba. Dorośli o tym zapominają i dlatego tworzą często bardzo nieefektywne systemy nauczania. Oczywiście miałem jakieś książki z przykładowymi programami w kodzie procesora 6510, a także „podglądałem” i analizowałem wiele gotowych programów, ale poczucie sukcesu dawało mi tylko to, że własnoręcznie napisany program wreszcie się nie zawiesił. W końcu zacząłem „czuć” ten procesor jak jazzmani „czują bluesa”. Język maszynowy składa się z prostych poleceń i był dla mnie nawet łatwiejszy do opanowania niż języki wyższego poziomu. Z łatwością „przesiadłem się” potem na procesory 8086/88 stosowane w pierwszych pecetach i napisałem kilka użytkowych programów maszynowych działających pod systemem operacyjnym DOS. Jednym z nich był „klawipol” – program do generacji polskich liter. DOS był bardzo przyjazny dla takich programów, pozwalał na niemal nieograniczony dostęp do portów i urządzeń. Straciłem zainteresowanie programowaniem w asemblerze dopiero, gdy nastał system Windows, ponieważ tam nie można było już swobodnie „buszować” procesorem po komputerze i w wielu przypadkach trzeba było używać narzuconych przez system poleceń API, a mnie konieczność stosowania się do wymysłów firmy Microsoft wcale nie bawiła. Ponadto zaczęło mnie to wszystko powoli przerażać. Cóż, byłem przecież tylko samoukiem.

Absolwent Wydziału Elektroniki, łatwo się domyślić, że próbował pan stworzyć jakiś projekt elektroniczny z komputerem. Co było tym pierwszym projektem?

Absolwent Wydziału Elektroniki, to prawda, ale równocześnie wieloletni pracownik Zakładu Teorii Obwodów Instytutu Telekomunikacji i Akustyki Politechniki Wrocławskiej, który z projektowaniem układów elektronicznych miał niewiele wspólnego. Wyroki losu

sprawiły, że prowadziłem ćwiczenia, seminary i laboratorium, a na Studium Wieczorowym również wykłady, z przedmiotu czysto teoretycznego, niezbyt zresztą lubianego przez studentów (a czasem nawet i przeze mnie), który wiązał się raczej ze specyficznymi zastosowaniami matematyki, a zwłaszcza algebry wyższej, niż z projektowaniem konkretnych układów elektronicznych. Elektroniką praktyczną zajmowałem się głównie w ramach tzw. „prac zleconych”, aby dorobić sobie do niewielkiej pensji podstawowej. Pracowałem między innymi przy konstruowaniu spektrometrów elektronowego rezonansu paramagnetycznego i byłem nawet współautorem jednego patentu.

W ramach swej działalności na uczelni napisałem wspólnie z kolegami tylko trzy prace związane z komputerem Commodore 64, w tym tylko jedna dotyczyła elektronicznego projektu współpracy przetwornika analogowo-cyfrowego z C64. Jednak na tym szybko się skończyło, bo moi przełożeni dali mi jasno do zrozumienia, że IBM brzmi znacznie poważniej niż CBM i jeśli chcę wiązać swą pracę naukową z komputerami, powinienem się zaprzyjaźnić raczej z wytworami tej pierwszej firmy. Dlatego też prawie wszystko co wiązało się z C64 było tylko moim prywatnym hobbym i zajmowałem się tym niemal wyłącznie w domu. Oprócz wspomnianego już programatora pamięci EPROM, zaprojektowałem również transoptorowy układ sprzęgający do USER PORT (to takie drugie, oprócz EXPANSION PORT ważne gniazdo w C64) oraz wykonałem kilka innych drobiazgów, jak pióro świetlne, łącze RS232 do PC, joysticki itp.

Skąd pomysł na takie projekty? Czy każdy projekt stworzony został w jakimś konkretnym celu?

Każde takie urządzenie miało swoje przeznaczenie. Transoptorowy układ sprzęgający miał na celu łączenie C64 przez USER PORT z różnymi urządzeniami, niekoniecznie dostosowanymi do tego portu pod względem elektrycznym, bez obawy uszkodzenia komputera (transoptor umożliwia całkowitą izolację elektryczną). C64 był wówczas dla mnie cennym nabytkiem i nie chciałem ryzykować jego zepsucia w trakcie przeprowadzanych eksperymentów. Miałem rozległe plany wykorzystania tego układu do różnych celów, ale bardzo niewiele zdążyłem zrealizować.

Pióro świetlne było wówczas modnym „gadżetem” działającym podobnie jak dzisiejsze „ekrany dotykowe”. Służyło do wybierania lub wskazywania różnych elementów wyświetlanych na ekranie, a także jako pióro do rysowania bezpośrednio po ekranie w programach graficznych. Jednak po jego wykonaniu okazało się, że jest dość niewygodne w użyciu, a nawet niekiedy za mało dokładne i dużo praktyczniej jest posługiwać się dżojstikiem; dlatego

odstawiłem je do lamusa. W przeciwieństwie do tego, łącze RS 232 do wymiany informacji między C64 i PC okazało się niezwykle użyteczne i wykorzystywałem je intensywnie w latach 90-tych przy programowaniu EPROMów szybkim, profesjonalnym programatorem sterowanym z peceta.

Proszę powiedzieć coś więcej o pańskim projekcie programatora EPROM. Czy jego konstrukcja była jakaś specjalna jak na lata 80te?

To nie była żadna nowatorska konstrukcja, ale na rynku polskim w owych latach takich programatorów sterowanych z C64 po prostu nie było i dlatego trzeba było go wykonać samodzielnie. Jednak sama idea programatora była dobrze znana i sposoby jego konstruowania były opisywane w literaturze elektronicznej. Obłożyłem się więc odpowiednimi książkami, a także fachowymi czasopismami i zaprojektowałem proste urządzenie, którego „sercem” był typowy i powszechnie wówczas dostępny układ scalony 8255 (programowalny element wejścia-wyjścia równoległego). Programator był połączony z C64 za pośrednictwem gniazda EXPANSION PORT, a ja musiałem napisać odpowiedni program sterujący do zapisu, odczytu i weryfikacji. Niestety, program ten nie zachował się, ale o ile pamiętam, menu ekranowe było napisane w BASIC’u, a procedury programowania, odczytu i weryfikacji – w języku maszynowym. Sam programator był umieszczony w standardowym, plastikowym pudełku o rozmiarach około 15x12x6 cm i – jak każdy prototyp – nie wyglądał zachwycająco pod względem estetyki i precyzji wykonania. Używał standardowych algorytmów programowania, dlatego nie działał zbyt szybko, ale po

usunięciu wszystkich błędów programowych stał się skuteczny i niezawodny. Obsługiwał pamięci wszystkich ówczesnych producentów o pojemności do 32 KB.

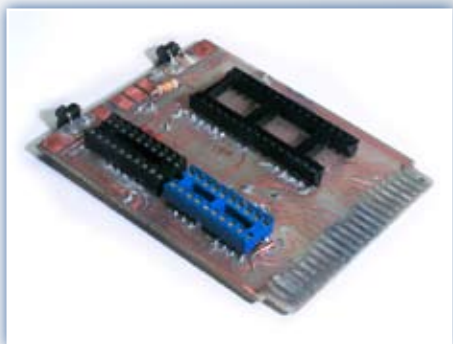
Żółta dioda elektroluminescencyjna to wskaźnik zasilania, zielona – sygnalizator odczytu, a czerwona – sygnalizator zapisu. Duży biały klawisz służył do przełączania między programowaniem i odczytem. Klawisze od 1 do 7 pozwalały na wstępny wybór typowych pamięci. Do zielonego gniazda można było wkładać w razie potrzeby klucze kodowe dla nietypowych pamięci. Czarne gniazdo służyło do połączenia odpowiednim kablem programatora z Commodore 64. W znajdującej się na górnej ścianie programatora dużej, białej podstawie z dźwignią zaciskową umieszczane były oczywiście EPROM’y.

Jak powstał programator pamięci EPROM to już wiadomo, że niedaleka droga budowy pierwszego kartridża. Nasuwa się pytanie jak powstał ten pierwszy kartridż?

Zanim powstał pierwszy kartridż miałem już trochę pomysłów na różne drobne usprawnienia programowe, przydatne dla C64. Jednak podstawowym problemem dla większości użytkowników tego komputera było to, że programy i gry były zwykle zapisywane na kasecie magnetofonowej i wczytywały się bardzo wolno. Istniał co prawda tzw. system TURBO, który przyspieszał odczyt dziesięciokrotnie, ale trzeba było go najpierw za każdym razem wczytywać i instalować w komputerze, co było niewygodne i w pewnym stopniu umniejszało jego zalety. Wpadłem więc na pomysł umieszczenia tego „przyspieszacza” w kartridżu, aby był dostępny przez cały czas od chwili urucho-

mienia komputera. Ponadto zamierzałem dołączyć do niego program umożliwiający regulację ustawiania głowicy magnetofonu, co było bardzo ważne dla unikania błędów zapisu i odczytu. Planowałem również wzbogacenie listy rozkazów C64 o różne dodatkowe dyrektywy ułatwiające przetwarzanie programów w języku BASIC, jak również przypisanie klawiszom funkcyjnym różnych często używanych poleceń. Wszystkie te moje zamysły połączyłem w jedną całość i w ten sposób powstał pewien system, który można już było wprowadzić do kartridża.

Konstruowanie kartridży rozpocząłem od zaprojektowania odpowiedniej płytki drukowanej, która w wersji testowej zamiast odpowiednich kości miała wlutowane podstawki, do których można było te kości (pamięci EPROM, bramki logiczne itp.) swobodnie wkładać i wyciągać.



Płytkę kartridża testowego Wnętrze Black Box’a V.4

Była ona umieszczona w standardowym pudełku do kartridży, w którym w miejscach gdzie były podstawki wycięto otwory. Teraz mogłem już zaprogramować EPROM, następnie włożyć go do odpowiedniej podstawki i rozpocząć testowanie kartridża. Po wykryciu błędu wystarczyło go usunąć z programu, EPROM wykasować, zaprogramować go ponownie tym razem już bez błędu i jeszcze raz umieścić w podstawie w celu ponownego przetestowania. Tylko kartridże do przeprowadzania prób miały podstawki. W egzemplarzach użytkowych wszystkie kości były wlutowane na stałe i miały one prostszą budowę. Niestety, pierwsze takie konstrukcje się nie zachowały; przedstawione na fotografiach płytki druko-



Programator pamięci EPROM do C64

wane są do nich podobne, ale dotyczą znacznie późniejszych wytworów.

Wszystkie główne prace, związane zarówno ze stroną programową projektu jak i techniczną realizacją kartridża wykonywałem sam; nikt mi w tym nie pomagał. Jednak jeden z moich kolegów, dr Czesław Michalik (obecnie docent Politechniki Wrocławskiej i kierownik Zakładu Teorii Obwodów) przez niemal cały czas mojej działalności cierpliwie testował moje kolejne kartridże, wyszukiwał błędy programowe i formalne, oraz inspirował do tworzenia nowych rozwiązań. On też wymyślił dla nich nazwę „Black Box”.

Początkowe próby konstruowania kartridża były niezbyt udane. Jednak w miarę usuwania błędów i ciągłego doskonalenia urządzenia powstawały coraz lepsze wersje tego kartridża, przy czym Black Box V.3 okazał się na tyle doskonały, że nadawał się już do codziennego użytku. Następnym z kolei udanym kartridżem był Black Box V.4, który został wzbogacony o kilka dodatkowych funkcji.

Praca nad tymi kartridżami nie była właściwie pracą w sensie jakiegoś mozołu czy wkładania w to wysiłku. Jak już wspominałem o tym wcześniej, była to dla mnie dobra zabawa i wszystko, co dotyczyło C64, robiłem dla własnej satysfakcji. Z każdym, kto się moimi działaniami interesował, dzieliłem się informacjami na temat kartridży, przekazywałem ich schematy elektryczne oraz oprogramowanie. Niewielką ilość gotowych kartridży rozprowadziłem wśród moich kolegów i znajomych prosząc o ocenę ich przydatności. Chciałem się również w ten sposób pochwalić swoimi dokonaniami. Wszystkie opinie były bardzo pozytywne, jednak o wartości komercyjnej tych kartridży przekonało mnie ostatecznie coś innego. Otóż po pewnym czasie okazało się ku mojemu wielkiemu zdumieniu, że ktoś zaczął je produkować i sprzedawać.

Z tego wynika, że kartridże Black Box v1 i v2 oficjalnie się nie ukazały i te co były w sprzedaży jako Black Box v1 i v2 to były podróbki wersji testowych?

Wersje Black Box V.1 i V.2 przedstawiały sobą tylko pewne etapy rozwojowe mojego pomysłu na kartridże i nie wyszły poza fazę prób. Przeznaczone były tylko do testowania i osobistego użytku. A używane były głównie przeze mnie i dr Czesława Michalika, który je sprawdzał, chociaż informacjami na temat tych kartridży dzieliłem się z każdym, kto o to poprosił. Niestety, nie zachowała się żadna dokumentacja ich dotycząca i mówiąc szczerze, nie pamiętam już dokładnie co zawierały. Wersja pierwsza na pewno musiała mieć wbudowany system TURBO i być może program do ustawiania głowicy. W każdej następnej wersji coś nowego dokładałem, poprawiałem lub usprawniałem. Wersja trzecia zawierała już wszystko co pierwotnie zamierzyłem, łącznie z

dodatkowymi rozkazami ułatwiającymi programowanie w języku BASIC oraz przypisaniem różnych poleceń klawiszom funkcyjnym. Poprawiłem w niej również wszystkie zauważone błędy i niedoróbki, które wykryliśmy. Nie wiem, czy te pierwsze wersje były kopiowane i sprzedawane. Może rzeczywiście tak było. Jednak zgodnie z tym co pamiętam, proceder ten ujawnił się dopiero poczynsz od wersji Black Box V.3. To właśnie najbardziej tą wersją chwaliłem się przed kolegami i znajomymi.

To od wersji Black Box v3 rozpoczął pan produkcję kartridży?

Tak, w 1991 roku specjalnie w tym celu założyłem małą, jednoosobową firmę ROM-BIT i rozpocząłem produkcję kartridży od wersji V.3, a wkrótce potem zacząłem wytwarzać również Black Box V.4, który w porównaniu z nieco wcześniej powstałym prototypem wzbogaciłem o ekranową prezentację tego kartridża. Pod koniec 1991 roku uzupełniłem swoją ofertę o nową wersję Black Box V.8.

Produkcję rozpocząłem od niewielkiej ilości sztuk, które były sprzedawane głównie przez zaprzyjaźnionego handlowca na giełdzie komputerowej i przez kilka sklepów. Przeciętny sklep zamawiał nie więcej niż kilka sztuk na miesiąc, więc żeby coś zarobić trzeba było pozyskać wielu nabywców, a ja nie miałem żadnego doświadczenia w dziedzinie reklamy i marketingu. Ogłosiłem się w Panoramie Firm i w Panelu Elektroniki, ale nie przyniosło to spodziewanych efektów. Chodziłem więc po sklepach i z różnym skutkiem proponowałem swoje wyroby. Niestety, te firmy, które wcześniej ode mnie rozpoczęły produkcję podróbek tych kartridży, już do pewnego stopnia opanowały rynek i wytwarzając ich znacznie więcej niż ja, dyktowały niskie ceny.

Jednak w 1992 roku popyt na te kartridże znacznie wzrósł i udało mi się pozyskać dwóch nowych, ważnych odbiorców: JTT Computer i Agencję Handlową KEN. Firmy te zamawiały zwykle jednorazowo po kilkadziesiąt sztuk kartridży, a w niektórych miesiącach (grudzień, styczeń, maj) zamówienia przekraczały czasem nawet setkę. W latach 1992/1993 zmuszony byłem do zatrudniania na umowę zlecenie kilku osób, bo sam nie dawałem sobie rady z montażem. Ale paradoksalnie ów wzrost popytu nie przyniósł spodziewanych dochodów, bo równocześnie znacznie wzrosła ilość tanich podróbek, więc aby utrzymać się na rynku, musiałem obniżyć ceny. Te podróbki były sprzedawane w całej Polsce i wciąż dochodziły do mnie sygnały, że w takim czy innym mieście widziano w sklepach jakieś kartridże Black Box z etykietką nie należącą do firmy ROM-BIT. Jednak nie wszyscy producenci kartridży decydowali się na taki proceder. Znalazły się dwie bardzo porządne firmy, ATRAX z Warszawy oraz MIAN z Wrocławia, które wykupiły ode mnie prawo do wytwarzania niektórych z nich. ATRAX produkował Black Box V.3, V.4 i V.8, zdaje się w dość dużych ilościach, a MIAN tylko dołączał niektóre moje systemy do swoich kartridży (np. system syntezy polskiej mowy).

W połowie 1993 roku zapotrzebowanie na kartridże zaczęło się powoli zmniejszać, a w 1994 roku obniżało się już dość szybko. W 1995 spadło poniżej poziomu opłacalności produkcji, co zmusiło mnie do likwidacji firmy.

Ile powstało wersji kartridża Black Box i czy kolejne wersje różniły się od siebie?

Znaczenie użytkowe i komercyjne miały przede wszystkim wersje V.3, V.4, V.8 i V.9. Istniały również wersje V.6 i V.7, ale zainte-



Wersje handlowe kartridży Black Box

resowanie nimi było niezbyt wielkie. Żaden taki kartridż nie dotrwał do dnia dzisiejszego, a ja obecnie nawet nie bardzo pamiętam, co te wersje zawierały. Zdaje się, że w wersji V.7 oprócz systemu Black Box był tam jakiś dodatkowy BASIC. Tego, co było w wersji V.6, nie przypominam sobie. Być może kartridż ten nie sprzedawał się dobrze lub uznałem go za wytwór nieudany i zaprzestałem produkcji. Pozostałe wersje, to jest pierwsza, druga i piąta, były to wersje przejściowe, które nigdy nie zostały przeze mnie wprowadzone na rynek. Nigdy też nie wypuszczałem wersji dwucyfrowych, jak np. 4.1 czy 8.2. Była to zwykła zmyłka podrabiającej moje kartridże konkurencji, ponieważ w środku nie było w nich żadnych zmian.

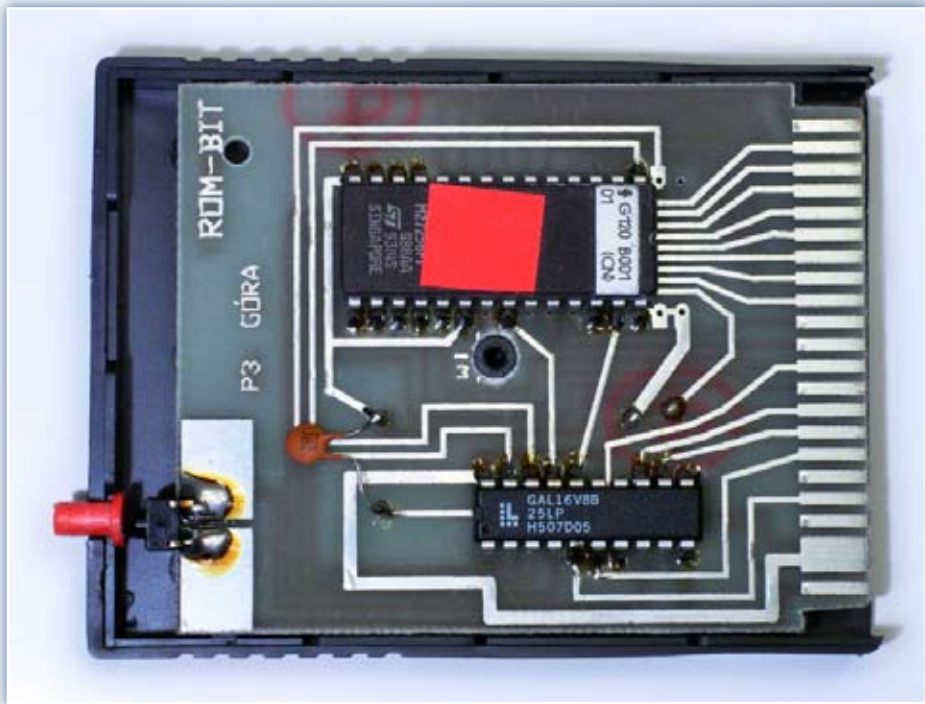
Wersje handlowe oczywiście różniły się od siebie. Miały również zróżnicowaną cenę, o której w praktyce decydowała pojemność użytej pamięci EPROM. Black Box V.3 był najprostszy i najtańszy, bo używał pamięć o pojemności tylko 8 kilobajtów. Black Box V.4 zawierał wszystko to, co oferowała wersja trzecia, a oprócz tego assembler-monitor i kilka innych dodatkowych rozkazów. Był nieco droższy, bo wykorzystywał EPROM 16-kilobajtowy.

Black Box V.8 był to kartridż dwusystemowy z 32-kilobajtową pamięcią. Oba systemy można było przełączać na kilka różnych sposobów. Jeden z tych systemów oparty był na wersji czwartej, został tylko wzbogacony o polecenie przełączania systemów. Drugi system oferował możliwość syntezy polskiej mowy i dźwięku. Był on pewną przeróbką programu SAM/RECITER, zgodnie z którą reguły angielskiej wymowy wymieniłem na dużo prostsze reguły polskiej wymowy. Z programu tego wykorzystywałem przede wszystkim cyfrowe próbki dźwięku i procedury generacji fonemów. Reguły polskiej wymowy opracowałem sam.

Black Box V.9, zawierający również 32-kilobajtową pamięć, został zbudowany pod koniec 1993 roku i był on ukoronowaniem całej mojej działalności związanej z projektowaniem kartridży oraz pełnym odzwierciedleniem wiedzy nabytej w wyniku ośmioletniej pracy z C64. Oferował 61 dodatkowych rozkazów. Zawierał nowy, znacznie lepszy program do ustawiania głowicy magnetofonu, nad którym pracowałem długi czas i udoskonalony system zapisu na taśmę TURBO. Nowością była też obsługa stacji dysków, a zwłaszcza szybkie wyświetlanie katalogu dyskowego i dodatkowe rozkazy dyskowe oraz system 15-krotnie przyspieszonego odczytu z dyskietki. Do dyspozycji były dwa zestawy klawiszy funkcyjnych i trzy przełączane ekrany tekstowe, przy czym na dwóch z nich dostępne były polskie litery. System obsługiwał drukarki z wejściem CENTRONICS, które mogły również drukować polskie litery. Kartridż oferował także specjalną minigrafikę utworzoną z połączonych ośmiu sprite'ów wykorzystującą pole o wymiarach 96 x 42 punkty.

Grafikę tę można było tworzyć w języku BASIC, animować, a także przesuwac za pomocą dżojstika. Black Box V.9 zawierał także nowy program realizujący polifoniczny instrument muzyczny z wieloma możliwościami ustawień i regulacji dźwięku. W czasie jego używania na ekranie była wyświetlana animowana klawiatura fortepianowa. W kartridżu tym umieściłem również rozbudowany system pomocy.

i kasowany, a ja zdrowo namieszałem w systemie, aby nie można było się łatwo zorientować, w którym momencie się uruchamiał i jak go ominąć. Ale nawet gdyby komuś to się udało, to kartridż bez właściwie zaprogramowanego układu GAL i tak nie chciał działać. Tego zabezpieczenia wówczas zdaje się nikt nie złamał (choć dziś nie stanowiłoby to problemu) i dlatego kartridż ten nie rozpowszechnił się



Black Box V.9 od środka

Jedną z najważniejszych cech tego kartridża było to, że miał on wbudowane sprzętowe zabezpieczenie przed podróbką. Zawierał specjalny układ scalony GAL16V8B, który stanowił serce tego zabezpieczenia. Jest to układ, którego struktura może być zaprogramowana elektrycznie (mniej więcej tak samo jak programuje się EPROM). To jest tak, jakby ktoś zamiast projektować i montować jakiś układ elektroniczny, miał już wszystkie potrzebne elementy umieszczone w jednej kości, a jego zadaniem byłoby tylko zaprogramować połączenia między tymi elementami. Tak działa ten GAL. Można w nim wykonać 2048 różnych połączeń, a liczba możliwych kombinacji tych połączeń jest astronomiczna. Układ daje się tak zaprogramować, aby nie można było odkryć, jakie połączenia zostały zrealizowane. Nawet gdy taką kość się rozetnie i ogląda pod mikroskopem, to i tak nic nie widać, bo te połączenia są realizowane czysto elektrycznie, a nie mechanicznie.

Zabezpieczenie kartridża polegało na tym, że po każdym restarcie systemu specjalny program sprawdzał czy GAL jest podłączony i odpowiednio zaprogramowany. Jeśli test wypadł negatywnie, to komputer się zawieszał. Program ten był zaszyfrowany. Po rozszyfrowaniu do pamięci RAM był on wykonywany

na całą Polskę, tak jak pozostałe. Niestety, powstał za późno, bo w 1994 roku zainteresowanie komputerem C64 zaczęło mocno spadać i popyt na kartridże zmniejszył się do tego stopnia, że doprowadziło to w następnym roku do zaprzestania ich produkcji. Tak więc najlepszy kartridż zdobył najmniejszą popularność. Cóż, tak czasem bywa.

Oprócz Black Box'ów powstały inne kartridże?

Tak, ale żaden nie przetrwał do dziś; nie mam już także żadnej dokumentacji ich dotyczącej i nie pamiętam szczegółów. Mogę jedynie powiedzieć, że kartridże te nie miały znaczącej wartości handlowej, bo cała produkcja opierała się głównie na Black Box'ach.

Czy Black Box był łączony z innymi kartridżami? Takie hybrydy kartridża Black Box i Final II lub X spotykało się na rynku. Były to produkcje Pana firmy?

Takie wersje hybrydowe niewątpliwie powstawały i prawdopodobnie jakaś część z nich była zaprojektowana przeze mnie. Niestety, nie zachowałem żadnych egzemplarzy tych kartridży, żadnej dokumentacji, nie pozostała też ani jedna instrukcja obsługi, nie mam więc żadnego punktu zaczepienia, w oparciu o któ-

ry mógłbym sobie coś próbować przypomnieć. A przecież od tego czasu minęło prawie dwadzieścia lat. O ile pamiętam, to takim dwusystemowym kartridżem był Black Box V.7. Co do pozostałych, jeśli w ogóle istniały, to niestety mam dziurę w pamięci, co tylko wskazuje na to, że nie odegrały one istotnej roli w mojej działalności, bowiem zwykle tylko rzeczy uznane za ważne dobrze się pamięta.

Wydawane przez pana kartridże miały na nalepce nazwę firmy i to chyba różniło je od podróbek?

Na pewno każdy Black Box wytwarzany przez firmę ROM-BIT miał na nalepce nazwę firmy. Firma ATRAX, która wykupiła prawo do wytwarzania Black Box'ów, też zdaje się umieszczała na nich napis ROM-BIT. Kartridże te różniły się jeszcze od podróbek tym, że do ich pamięci EPROM wpisywane było kilkakrotnie moje nazwisko i inne dane osobowe, a w podróbkach często je usuwano, chociaż bywało i tak, że kopiowano zawartość tej pamięci bajt po bajcie bez żadnych zmian. W kartridżu Black Box V.8 jeden z takich zapisów był ukryty i można było go ujawnić dopiero poprzez przytrzymanie klawisza „1” (jeden) i włączenie zasilania komputera lub wykonanie resetu – wtedy wyświetlał się na ekranie. Chyba nikt się nie zorientował, że tak jest; przynajmniej w żadnej wówczas testowanej przeze mnie podróbce tego zabezpieczenia nie usunięto.

Były plany stworzenia Black Box'a v10, czy może zupełnie nowego kartridża?

Black Box V.9 był ostatnim kartridżem, jaki zaprojektowałem. Gdyby popularność C64 nie spadła, na pewno by się na tym kartridżu nie skończyło, ale stało się inaczej.

Nie myślał pan przenieść produkcji na inny komputer np. Amiga czy PC?

Zawsze uważałem siebie bardziej za twórcę niż za wytwórcę, dlatego nigdy nie planowałem kariery producenta czy biznesmena. Firma ROM-BIT powstała w ściśle określonym celu i gdy realizacja tego celu dobiegła końca, zakończyłem swoją działalność. Owszem, myślałem o zaprojektowaniu urządzenia realizującego syntezę polskiej mowy w postaci zewnętrznej przystawki lub karty do PC. Chciałem w tym celu wykorzystać jeden z dostępnych mikrokontrolerów jednoukładowych. Jednak przeceniłem możliwości realizacji tego pomysłu, ponieważ to wymagałoby działania w kilkucosobowym zespole, którego nie miałem, albo co najmniej kilku lat samotnej pracy. Poza tym wkrótce okazało się, że ktoś takie urządzenie zaczął już produkować, a ja chciałem stworzyć coś zupełnie nowego. Uznałem więc, że nie ma szans na realizację moich planów i szybko z nich zrezygnowałem.

W wywiadzie z panem Waldemarem Czajkowskim przeczytałem, że stworzył pan urządzenie do kopiowania kaset magnetofonowych, do którego można podpiąć kilka magnetofonów. Mógłby pan coś więcej o tym powiedzieć?

Waldemar Czajkowski to właśnie ten „za przyjaźniony handlowiec”, o którym wspominałem wcześniej i który od samego początku sprzedawał moje kartridże. Poznaliśmy się na uczelni w czasie gdy był studentem Wydziału Elektroniki Politechniki Wrocławskiej. Prowadził studio komputerowe, którego działalność była oparta głównie na kopiowaniu i sprzedawaniu gier do komputerów domowych, a zwłaszcza do C64. Komputer ten był, jak na owe czasy, dość drogi, a stacja dysków – jeszcze droższa. Dlatego dla wielu jego użytkowników głównym urządzeniem do wczytywania gier był magnetofon zwany DATASSETTE, a głównym nośnikiem były kasety. Ponieważ C64 miał tylko jedno złącze dla magnetofonu kasetowego więc za pomocą jednego komputera można było w danym czasie nagrywać tylko jedną kasetę.

Waldemar Czajkowski zwrócił się do mnie z pytaniem, czy jest możliwe skonstruowanie urządzenia, które by zwielokrotniało liczbę tych złącz, tak aby z jednego komputera można było nagrywać jednocześnie więcej kaset. Ponieważ nie widziałem w tym żadnego problemu, przyjąłem zamówienie na kilka takich urządzeń. Nazywaliśmy je „nagrywarkami”, chociaż tak naprawdę one niczego same nie nagrywały, a tylko umożliwiały podłączenie do każdej z nich ośmiu magnetofonów. Nagrywarka taka była podłączona kablem do odpowiedniego złącza C64, a na swojej obudowie miała osiem złącz do DATASSETTE, dokładnie takich samych jak w komputerze. Liczba tych złącz teoretycznie mogła być nieograniczona, a w praktyce było ich tyle, ile zdołało się zmieścić na ściankach standardowej obudowy. Tak więc za pomocą jednego komputera i jednej nagrywarki można było zapisywać jednocześnie na ośmiu kasetach. Wykorzystując więcej nagrywarek, można było zwielokrotnić liczbę nagrywanych kaset, łącząc je kaskadowo lub szeregowo. Po prostu, zamiast podłączać magnetofony do złącz wyjściowych urządzenia, można było do niektórych z nich, a nawet do wszystkich, podłączyć inne nagrywarki, co powodowało „rozmnożenie” liczby takich złącz.

Zdaje się, że urządzenie to bardzo się Waldemarowi Czajkowskiemu spodobało, w każdym razie zamawiał je potem jeszcze kilkakrotnie. Wytworzyłem ich w sumie chyba kilkanaście sztuk, a może nawet jeszcze więcej, dziś już nie pamiętam. Jako produkt na indywidualne zamówienie, było dość drogie, ale przypuszczam, że jego koszty zwracały się po jednym dniu intensywnego używania.

Jak pan obecnie się zapatruje na to, że ludzie jeszcze coś robią na komputerze C64 i nadal go utrzymują przy życiu?

To wydaje się być zadziwiające w epoce pecetów z wielordzeniowymi procesorami, kilkusetgigabajtowych dysków twardych czy PlayStation 3 z gramami o jakości HD. Jednak z drugiej strony nie ma się czemu dziwić, jeśli zważyć, że dla tych, którzy poprzednio żyli w epoce „przedinformatycznej” ten pierwszy komputer domowy był jak pierwsza miłość, którą zawsze się mile wspomina. Był on jak na owe czasy bardzo wszechstronny, oferował mnóstwo programów użytkowych, które potem były przenoszone na PC. Pamiętam programy graficzne (jeden z takich bardziej znanych nazywał się chyba GEOS), arkusze kalkulacyjne, a także programy do rysowania schematów elektrycznych. Istniał nawet program do projektowania płytek drukowanych. Pierwsze płytki do kartridży projektowałem właśnie na Commodore, następnie drukowałem rysunki w skali 2:1 i zanosilem do zakładu, który je wykonywał. Jako że C64 miał wbudowany mały syntezator dźwięku, tworzone do niego nawet edytory muzyczne. Ja na jednym z nich (nazywał się chyba FINALE), próbowałem aranżować różne utwory muzyczne, a nawet napisałem kilka własnych kompozycji.

Jednak większość użytkowników C64 używało go głównie do gier, które miały swój niepowtarzalny urok, jakiego nie ma wiele gier współczesnych. Fantastyczny świat pokazany za pomocą prostej grafiki wspomaganej ośmioma małymi „sprite’ami” miał, i prawdopodobnie wciąż ma, większą siłę oddziaływania na wyobraźnię niż pokazane w wysokiej rozdzielczości, do bólu realistyczne postacie współczesnych gier. Poza tym tych gier na C64 było bardzo dużo, chyba ponad dziesięć tysięcy i były łatwo dostępne, często na drodze wymiany. A te, które trzeba było sobie kupić, były tanie, bo przecież te ówczesne kilkadziesiąt tysięcy, jakie należało zapłacić za kasetę z gramami, to kilka dzisiejszych złotych. Dziś, patrząc na cennik gier do PS3, może się w ogóle odechcieć grać. Ponadto niektórych ludzi, takie współczesne gry mniej wciągają niż te na C64, a są nawet tacy, co uważają je wręcz za „męczące”.

Nie dziwię się więc, że są osoby wciąż wierne C64, chociaż w najśmielszych swoich przypuszczeniach nie przewidziałem, że to zjawisko będzie miało aż taką skalę. Po likwidacji firmy sprzedałem swoją nowszą wersję Commodore i stację dysków, a wszelkiej dokumentacji i oprogramowania się pozbyłem. Gdybym przewidział ten renesans C64, to na pewno bym tego nie uczynił i miałbym dzisiaj znacznie więcej do powiedzenia na temat tego komputera.

Nie myślał pan, żeby tak w wolnych chwilach stworzyć coś jeszcze do C64, może nowy kartridż?



R. Drahokaupil pod Śnieżką w 2005 r.

Nie i to z wielu powodów. Po pierwsze, od 1995 roku, po likwidacji firmy, przestałem się interesować elektroniką i komputerami. Zająłem się zupełnie innymi sprawami, na które poprzednio nie miałem czasu. Po drugie, nawet gdybym chciał coś zrobić, to i tak nie mam już stacji dysków, a nawet DATASSETTE, nie mówiąc już o odpowiednich programach i dokumentacji swoich dawnych prac. Po trzecie, prawie wszystkiego, co dotyczy C64, już zapomniałem, a przypominanie sobie tego byłoby bardzo pracochłonne i zajęłoby mi zbyt wiele czasu, na co nie mogę sobie pozwolić. Po czwarte, moją dewizą życiową jest to, aby spoglądać w przyszłość, a nie oglądać się za siebie, więc wolę się zająć czymś dla mnie zupełnie nowym, niż wracać do starego.

A czym się pan zajmuje w wolnych chwilach, ma pan jakieś hobby, zainteresowania?

Od dzieciństwa interesuję się muzyką i fotografią. Pierwszy aparat fotograficzny otrzymałem w prezencie od rodziców już w wieku kilkunastu lat. Był to kupiony od rosyjskiego żołnierza „Zorkij S”, będący wierną kopią przedwojennego niemieckiego aparatu „Leica”. Prawie zawsze sam zajmowałem się wywoływaniem filmów i wykonywaniem odbitek na papierze fotograficznym, początkowo czarno-białych, a potem również kolorowych. Kilka razy zmieniałem markę aparatu, a od pewnego czasu posługuję się aparatem cyfrowym, gdzie oczywiście cała chemia jest niepotrzebna. Fotografuję dla przyjemności, głównie podczas wakacyjnych wyjazdów. Fotograficzne hobby łączę z zamiłowaniem do lasu i gór; większość swoich zdjęć wykonałem w Sudetach.

Zainteresowanie muzyką też zaczęło się w wieku wczesnoszkolnym, ponieważ równolegle z nauką w zwykłej szkole podstawowej, uczęszczałem do podstawowej szkoły muzycz-

nej, ucząc się gry na fortepianie. Po ukończeniu sześciu klas tej szkoły przerwałem naukę i przestałem ćwiczyć. Dziś już niemal nic mi z tego nie zostało, poza pewnym zamiłowaniem do muzyki. Lubię słuchać wybranych kompozytorów muzyki poważnej, cenię sobie piękne arie operowe i operetkowe. Szanuję każdy gatunek muzyczny pod warunkiem, że jego celem jest coś więcej niż robienie hałasu, ale szczególnie preferuję te utwory, w których występują nie zdominowane przez rytmikę piękne melodie i konsonansowe akordy. Do czynnego uprawiania muzyki wróciłem na krótko gdy nastąpiła era komputerów i powstały edytory „midi”, ponieważ wówczas nie musiałem grać na żadnym instrumencie; wystarczyło „ułożyć” muzykę, a grała za mnie maszyna. Pierwszym takim moim muzycznym komputerem był C64, a potem PC wzbogacony o kartę „Wave Blaster”. Obecnie skupiam się na słuchaniu i gromadzeniu nagrań audio i wideo koncertów, oper, operetek i musicali z dźwiękiem dobrej jakości, np. DTS.

W drugiej połowie lat siedemdziesiątych ubiegłego wieku kolejnym moim hobby stało się filmowanie, kiedy to będąc służbowo w ZSRR jako opiekun grupy studentów kupiłem tam sobie kamerę „Kwarc” na film „super 8”. Była to zabawa wymagająca wiele cierpliwości i samozaparcia, gdyż trzeba było za pomocą specjalnej sklejkarki łączyć poszczególne odcinki filmu, a te sklejkarki się często rwały w trakcie jego wyświetlania. Swoje filmowe zainteresowania kontynuowałem w późniejszych latach, używając videokamery VHS-C. Obecnie mam kamerę cyfrową Panasonic NV-GS75EP, a program Pinnacle Studio 11 Ultimate czyni edycję i montaż filmów rzeczą łatwą i przyjemną. Tworzę typowe filmy amatorskie, pamiątki rodzinne, filmy z wakacji, sprawozdania z wycieczek, itp. Nakręciłem również film o swoim



Z kamerą Kwarc w 1978 r.

mieście. Pokazuję je wyłącznie rodzinie, przyjaciółom i dobrym znajomym.

Cóż jeszcze? Lubię majsterkowanie. Ponadto interesuję się również zagadnieniami społecznymi i moralnymi, religioznawstwem, wybranymi zagadnieniami filozofii i psychologii, a nawet, do pewnego stopnia, parapsychologią i ezoteryką. Badam też pochodzenie i rozważam znaczenie różnych słów. Próbuję spisywać swoje myśli. I to by było wszystko.

Dziękuję za rozmowę i życzę wszystkiego najlepszego w tym roku !!!

Ja także dziękuję, ponieważ ta rozmowa pomogła mi w przypomnieniu sobie wielu przeszłych faktów i zdarzeń, których być może bez niej nigdy bym sobie nie przypomniał. Życzę również wszystkiego najlepszego całemu zespołowi redakcyjnemu i wszystkim czytelnikom C&A Fan.

Wywiad przeprowadził: Ramos



Krótką historia kartridży Black Box

Jak wiadomo, komputery z ośmiobitowymi procesorami zaczęły się pojawiać w krajach zachodnich w latach osiemdziesiątych XX wieku, C-64 wypuszczono na rynek w 1982 roku, a PC XT sprzedawano już nawet od 1981 roku. Jednak było to dość duże pudło i w dodatku drogie, więc większą popularność zdobyły mniejsze i tańsze komputerki domowe firmy Commodore (C-64, C-128), Sinclair (ZX Spectrum), Atari i Apple. Niestety, Polska była wtedy w stanie wojennym, środki masowego przekazu, takie jak prasa, radio i telewizja (Internetu nie było) o tych nowościach słabo informowały, trudno było wydostać się poza obszar krajów socjalistycznych i większość Polaków początkowo nawet nie wiedziała co się dzieje. Dopiero od połowy lat osiemdziesiątych zaczęły przenikać do kraju bardziej obszerne informacje o komputerach osobistych. Im bliżej było do Okrągłego Stołu, tym łatwiej było wyjechać na Zachód i coraz więcej ludzi przywoziło z zagranicy ZX Spectrum lub C-64, które były wtedy najbardziej popularne. Jednak w polskich sklepach ani tych mikrokomputerów, ani legalnego oprogramowania nabyć nie można było i wszystko opierało się na prywatnym imporcie oraz nieautoryzowanych kopiach programów.

Ja swój pierwszy C-64 zamówiłem chyba około roku 1985, w zachodnioniemieckiej firmie handlującej elektronicznymi instrumentami muzycznymi (jej adres dostałem od jednego z kolegów), dokonując przelewu bankowego i czekając kilka miesięcy na dostawę. Dlaczego C-64? Był to raczej bardziej zbieg okoliczności niż świadomy wybór. Trochę oparłem się również na dobrych radach bardziej zorientowanych kolegów. Muszę tutaj zaznaczyć, że nie jestem informatykiem, a na Wydziale Elektroniki Politechniki Wrocławskiej, na którym studiowałem, nie było wówczas takiego przedmiotu. Nie miałem więc zielonego pojęcia o tym, jak działa komputer i wszystkie informacje na ten temat musiałem zdobywać

sam. Najpierw poznałem standardowy język BASIC i zacząłem pisać proste programy w tym języku, które działały dość wolno, bo były oparte na zasadzie interpretera. Zaopatrzyłem się więc w różne assemblyery oraz monitory i zacząłem eksperymentować z programowaniem w języku maszynowym. Dużą pomocą była dla mnie książka (tytuł odtwarzam z pamięci) „Commodore Programmers Reference Guide”, którą nazywano „biblią” Commodore i która zawierała absolutnie wszystkie informacje potrzebne do tego, aby rozgryźć tę maszynę. Był tam zarówno dokładnie opisany język BASIC, jak i wszystkie rozkazy procesora 6510. Dokładnie omówiono w tej książce całą mapę pamięci C-64, wszystkie ważne adresy portów i metody podłączania do komputera pamięci EPROM umieszczanej w kartridżach. Wiele trzeba było samemu wydedukować, ale wspomniana książka w połączeniu ze schematem elektronicznym C-64 stanowiła zupełnie wystarczającą podstawę do opanowania programowania na tym komputerze, poznania jego tajników i konstrukcji własnego kartridża.

Zaintrygowany programem SAM/RECITER, który realizował syntezę angielskiej mowy, zacząłem się zastanawiać, czy dałoby się stworzyć program mówiący po polsku. Po wielu miesiącach prób udało się w końcu otrzymać wersję, która w sposób w miarę zadowalający (jak na owe czasy) realizowała syntezę polskiej mowy. Oczywiście, nie napisałem zupełnie nowego programu. Wykorzystałem sposób generacji elementarnych dźwięków mowy (fonemów) z programu SAM/RECITER. Natomiast całkowicie zmieniłem reguły mowy, wprowadzając polskie zasady zamiast angielskich. W sumie te polskie reguły okazały się dużo prostsze od angielskich, ponieważ w naszym języku pisownia jednoznacznie określa wymowę (z niewielkimi wyjątkami), a w języku angielskim jest dużo niejednoznaczności. Program ten został zademonstrowany we wrocławskim Klubie Dziennikarza już pod koniec 1985 roku.

W miarę jak moje wysiłki w kierunku opanowania C-64 stawały się coraz bardziej skuteczne, otrzymywałem coraz ciekawsze rezultaty swojej pracy, co sprawiało mi wiele satysfakcji. Napisałem wiele oddzielnych programów, które potem postanowiłem połączyć w jedną całość, konstruując swój pierwszy kartridż. Z uwagi na elektroniczne wykształcenie nie miałem problemów z zaprojektowaniem i wykonaniem płytki drukowanej. Musiałem również sam zbudować sobie programator pamięci EPROM i napisać program sterujący. Pierwsze próby nie były zbyt udane i dopiero trzecia wersja kartridża okazała się zadowalająca. Jego idea, jak również wszystkich późniejszych wersji, polegała na wzbogaceniu interpretera BASIC o nowe, użyteczne rozkazy. W ten sposób powstał pierwszy Black Box V.3. Niedługo potem zaprojektowałem następną wersję – Black Box V.4, która została wzbogacona o kilka dodatkowych funkcji.

C-64 rozpracowywałem i pisałem programy w języku asemblera całkiem sam, nikt mi w tym nie pomagał. Jednak jeden z moich kolegów, dr Czesław Michalik (obecnie docent Politechniki Wrocławskiej i kierownik Zakładu Teorii Obwodów) przez cały czas mojej działalności cierpliwie testował moje kolejne kartridże, wyszukiwał błędy programowe i formalne oraz inspirował do tworzenia nowych rozwiązań. On też wymyślił nazwę „Black Box”.

Rozgryzanie C-64, doskonalenie sztuki programowania i tworzenie kolejnych kartridży sprawiało mi dużą przyjemność. Zupełnie nie przychodziło mi wówczas do głowy, że mogą one mieć jakąś wartość handlową; wystarczała mi zupełnie satysfakcja z ich konstruowania. Jednak w końcu niewielką ilość tych kartridży rozprowadziłem pośród kolegów i znajomych z prośbą o ocenę ich przydatności. Nie ukrywam, że chciałem się też w ten sposób pochwalić swoimi dokonaniem. Jakież było moje zdumienie, gdy po pewnym czasie stwierdziłem, że te kartridże ktoś zaczął

wytwarzać i sprzedawać. Dopiero to mnie ostatecznie przekonało, że istnieje na nie wystarczająco duży popyt, aby pomyśleć o podjęciu ich produkcji, tym bardziej, że C-64 zaczęły się już pojawiać w polskich sklepach komputerowych. W 1991 roku założyłem małą firmę jednoosobową ROM-BIT, zainwestowałem w PC-ta oraz porządny programator EPROM-ów i rozpocząłem produkcję. Równocześnie pracowałem nad nowymi kartridżami, czego owocem była udana wersja Black Box V.8, zawierająca opracowany w latach wcześniejszych program syntezy polskiej mowy. W 1992 roku popyt na kartridże do C-64 osiągnął na tyle wysoki poziom, że dwie firmy, MIAN z Wrocławia i ATRAX z Warszawy, wykupiły ode mnie prawo do wytwarzania niektórych z nich. Jednocześnie zacząłem otrzymywać zamówienia od poważnych biur handlowych i hurtowni, jak Agencja Handlowa KEN czy JTT Computer. Zmuszony byłem także do zatrudniania na umowę-zlecenie kilku innych osób, bo sam nie dawałem sobie rady z montażem urządzeń. Jednak nie osiągnąłem zbyt dużego dochodu, ponieważ istniały również firmy, które produkowały tych kartridży znacznie więcej i taniej niż ja, wcale nie pytając mnie o pozwolenie. Z jedną taką się nawet procesowałem, ale proces trwał kilka lat i nie przyniósł oczekiwanych rezultatów. Zatem przez taką „konkurencję” byłem zmuszony do znacznego obniżania cen, co zmniejszało realne zyski. Szczyt popularności Black Box-ów

nastąpił na przełomie lat 1992/1993, po czym popyt zaczął powoli spadać. W 1994 roku zaczął zmniejszać się już dość gwałtownie (i nie pomogło wprowadzenie nowej, chyba najbardziej profesjonalnie wykonanej wersji V.9), a w następnym roku spadł niemal do zera, co zmusiło mnie do likwidacji firmy. Jak na ironię, sąd uznał moje osobiste prawa do Black Box-ów dopiero w 1995 roku i nakazał pozwanej firmie zaprzestanie ich produkcji, gdy była to już przysłowiowa „musztarda po obiedzie”.

Trzeba również powiedzieć o problemie legalności oprogramowania na C-64. Otóż w tamtych czasach prawie nie było na rynku polskim autoryzowanych programów. Nie tylko na giełdach komputerowych, ale również i w sklepach sprzedawane były masowo powielane kopie gier do C-64, które wedle dzisiejszej miary byłyby uznane za „pirackie”. Niektórzy się na nich nawet nieźle dorobili, zaczynając od handlu na giełdach, a kończąc na otwieraniu sieci sklepów. Prawo ich za taki proceder nie ściagało. Ten stan rzeczy był dość powszechnie akceptowany, chociaż pojawiały się czasem pojedyncze głosy sprzeciwu. Dopiero uchwalone w 1994 roku nowe prawo autorskie dotyczyło również oprogramowania. Było ono jednak łagodniejsze niż obecne, znowelizowane. Przewidywało pewien okres przejściowy i zaczęło w pełni obowiązywać dopiero od 1995 roku. Ponadto wszyscy, którzy już byli w posiadaniu

nielegalnego oprogramowania, mogli je nadal używać, nawet po wejściu nowej ustawy.

Jak widać, przed 1995 rokiem w kwestii oprogramowania wszystko było możliwe. Jednak obecnie nie można tego surowo oceniać i trzeba na to spojrzeć z wyrozumiałością. Nie możemy osądzać tych ludzi z dzisiejszej perspektywy, tak jak nie możemy osądzać np. św. Tomasza z Akwinu, że opowiadał się za karą śmierci dla heretyków. Takie były czasy. Ja wystąpiłem do sądu przeciw pewnemu wrocławskiemu przedsiębiorcy tylko dlatego, że uświadomiłem sobie, a właściwie uświadomili mi to moi koledzy i znajomi, że ktoś robi wielkie pieniądze na tym, co było owocem mojej kilkuletniej, intensywnej pracy, mimo że zwrócono mu uwagę, iż narusza czyjeś prawa. Jednak z dzisiejszej perspektywy patrzę na to dużo łagodniej i nie mam już żadnej urazy do tego pana. Sam pod tym względem też nie byłem nieskazitelny, bo chociaż wszystkie rozwiązania techniczne i większość rozwiązań programowych to moje oryginalne pomysły, to jednak opierałem się również na cudzych programach, np. wykorzystałem procedury generacji fonemów oraz próbki fonemów z programu SAM/RECITER. Dziś wiem, że w owym początkowym okresie przechodzenia od mentalności socjalistycznej do kapitalistycznej, taki rozwój wydarzeń był nieunikniony.

Romuald Drahokaupil



R. Drahokaupil pod Śnieżką w 2005 r.

Jak podłączyć kartridż do C-64 i jak przełączać pamięć?

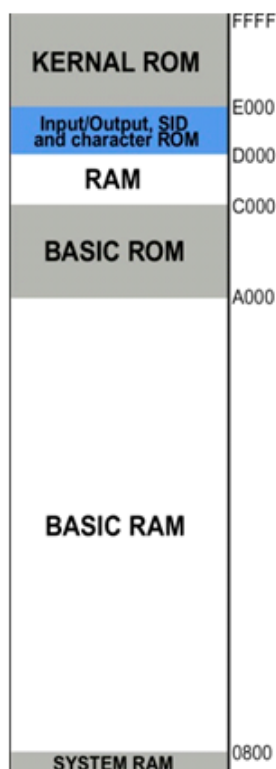
Słowo kartridż pochodzi od angielskiego wyrazu „cartridge” i znaczy dosłownie „nabój”. Idea kartridża jest owocem dążenia producentów różnych urządzeń komputerowych do tego, by ich wytwór nigdy nie był układem zamkniętym, w którym nic już nie można zmienić. W pecetach takim przejawem zapewnienia możliwości rozszerzania systemu o nowe funkcje są umieszczane na płycie głównej tzw. „gniazda rozszerzeń”, do których można wkładać dodatkowe karty o różnym przeznaczeniu, np. karty dźwiękowe, graficzne, sieciowe, modemy itp. W małych maszynach klasy C64 nie dało się tego zrobić w taki sposób, dlatego wymyślono tzw. „port rozszerzeń”, do którego można wtykać od zewnątrz różne dodatkowe urządzenia zwiększające możliwości tych komputerów. Do idei zewnętrznych portów rozszerzeń wrócono zresztą potem w laptopach i notebookach; przykładem tego jest gniazdo ExpressCard, w którym można umieszczać dodatkowe elementy współpracujące z tymi komputerami.

C64 CARTRIDGE/EXPANSION PORT jest jednym z wcześniejszych takich rozwiązań, które pozwoliło na znaczne zwiększenie możliwości Commodore 64. Poprzez ten port można było dołączać różne urządzenia zewnętrzne współpracujące z tą maszyną. Jednym z takich możliwych do podłączenia „gadżetów” był kartridż – małe pudełko ze stykami po jednej stronie. Umieszczenie w tym pudełku dodatkowej pamięci EPROM z załadowanymi do niej różnymi programami użytkowymi czy grami umożliwiała uruchamianie ich z poziomu C64. Dzięki kartridżowi można było też wzbogacić interpreter BASIC o dodatkowe rozkazy. Ale kartridż nie był tylko takim dawnym odpowiednikiem współczesnej pamięci USB. Za pomocą kartridża można było kompletnie zmienić system operacyjny, a nawet wymienić procesor.

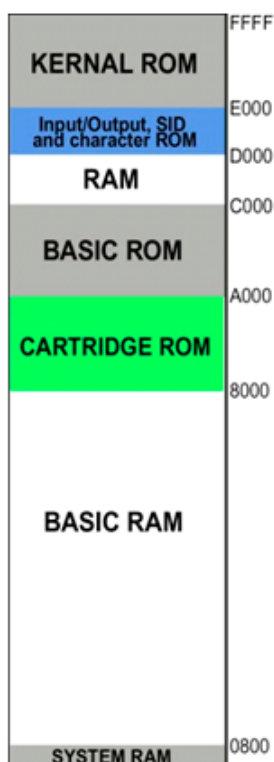
Obecnie postaram się wyjaśnić, jak podłączyć kartridż z dodatkową pamięcią EPROM do komputera i jak sprawić, by ta pamięć EPROM pojawiała się na mapie pamięci C-64 i z niej znikła. Niestety, materiałów, w których

było wszystko o kartridżach do C-64, już dawno pozbyłem się, więc musiałem pewne informacje odtworzyć z własnej pamięci, a ludzka pamięć, jak wiadomo, jest zawodna. Jednak spróbuję jakoś to przedstawić. Ponieważ procesor 6510 ma 16-bitową szynę adresową, obejmuje 64 KB pamięci. Tyle też umieszczono w C-64 RAM-u. Jednak pamięć RAM jest ulotna, więc system operacyjny trzeba było umieścić w 8-kilobajtowej pamięci stałej ROM, którą projektanci C-64 nazwali KERNAL (jądro). Tak samo w 8-kilobajtowej pamięci ROM należało umieścić interpreter BASIC. Ponadto procesor powinien mieć dostęp do rejestrów portów obsługujących wejścia i wyjścia oraz rejestrów generatora dźwięków SID, więc muszą być one dla niego „widoczne” w odpowiednim obszarze adresowym. Dostępne muszą być również mapy bitowe znaków graficznych. Mapa pamięci C-64 w wersji podstawowej wygląda więc tak, jak na rys. 1.

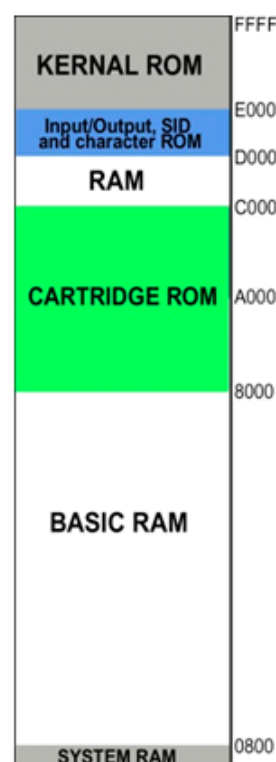
Oczywiście C-64 ma pełne 64KB RAM-u, ale w tej konfiguracji pamięć w obszarze adresowym BASIC ROM i KERNAL ROM jest



Rys. 1

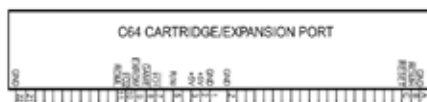


Rys. 2



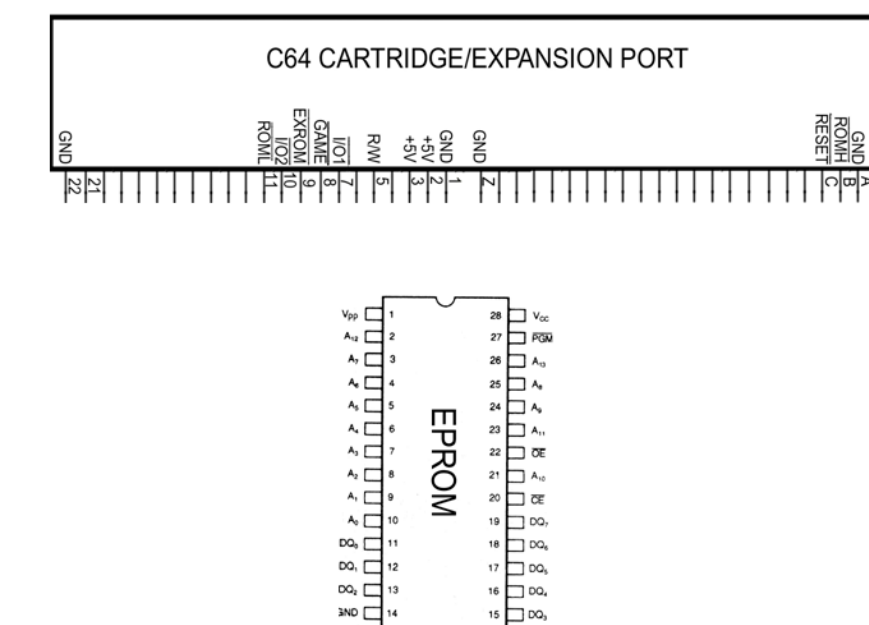
Rys. 3

dostępna tylko dla zapisu. Próba odczytania jakiegokolwiek bajtu pamięci z tego obszaru zawsze prowadzi do odczytania ROM-u. RAM w \$A000-\$BFFF i \$D000-\$FFFF nie może być więc użyty. Jednak rezygnacja z wykorzystywania RAM-u w tych lokalizacjach byłaby marnotrawstwem, ponieważ RAM-u nigdy nie jest za dużo. Dlatego producent przewidział możliwość programowego „wyłączenia” BASIC ROM i KERNAL ROM, poprzez ustawienie odpowiedniej konfiguracji bitów w jednej z komórek pamięci zarezerwowanej dla systemu (od \$0000 do \$07FF). Jest to dokładnie adres \$0001. Wyzerowanie najmłodszego bitu liczby zapisanej pod tym adresem usuwa z mapy pamięć BASIC ROM, a wyzerowanie następnego bitu usuwa również KERNAL. Wyłączając KERNAL trzeba jednak uważać, ponieważ obsługuje on przerwania i przed jego usunięciem z mapy pamięci zawsze trzeba najpierw wyłączyć te przerwania, bo w przeciwnym razie system się zawiesi.



Rys. 4

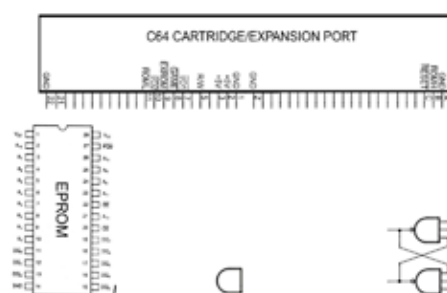
Jeśli chodzi o CARTRIDGE ROM (w kartridżach zwykle stosowano pamięć EPROM), to nie można go włączać i wyłączać w opisany wyżej sposób. Jedyna metoda, którą można zastosować, polega na oddziaływaniu na komputer przez sam kartridż podłączony do gniazda EXPANSION PORT. W tym gnieździe, przedstawionym na rys. 4, są dwa zestyki, EXROM (pin 9) i GAME (pin 8); mają one status wejść. Jeżeli do tych wejść nic nie jest podłączone – a jest tak, gdy komputer pracuje bez kartridża – to jest tak, jakby na nich panował stan wysoki H i wówczas mapa pamięci przedstawia się tak, jak na rys. 1. Jeśli na zestyku EXROM wymusi stan niski L (np. przyłączymy go do masy), wówczas konfiguracja się zmienia i mapa pamięci wygląda tak, jak na rys. 2. Możemy wtedy przyłączyć do portu 8 KB EPROM. Jeśli wywołamy stany niskie na obu wejściach, EXROM = L i GAME = L, to, jeśli dobrze pamiętam, konfiguracja pamięci jest taka, jak na rys. 3. Można wtedy włączyć do systemu 16 KB EPROM, jednak BASIC ROM jest wówczas wyłączony. Jest jeszcze trzecia możliwość, że EXROM = H i GAME = L, która prowadzi do umieszczenia CARTRIDGE ROM w obszarze KERNALA, ale ja jej nigdy nie wykorzystywałem, bo ta konfiguracja wyłącza całkowicie system operacyjny C-64, a komputer jest wtedy pod wyłączną kontrolą kartridża. Ten sposób był stosowany do podłączania niektórych profesjonalnych kartridży z gramami. T a k więc podłączenie kartridża zgodnie z konfiguracją pokazaną na rys. 2 wygląda następująco:



Rys. 5

Żeby nie zaciemniać sprawy, na tym rysunku pominięto podłączenie do EPROM-u zasilania, masy, szyny adresowej i szyny danych. Wyjście ROML, przyjmując stan niski, aktywizuje EPROM za każdym razem, gdy tylko procesor wykonuje jakąś operację w przedziale adresowym \$8000-\$9FFF. Tak oczywiście musi być, bo w systemie komputerowym, w którym wiele układów pamięci jest podłączonych do wspólnej szyny danych, tylko jeden układ w danym czasie może być aktywny. Jest to najprostszy, ale najmniej doskonały sposób podłączenia kartridża do C-64, bo RAM w obszarze adresowym \$8000-\$9FFF w ogóle nie może być wówczas wykorzystany.

Najlepiej by więc było, aby można było swobodnie przełączać konfiguracje pamięci z rysunków 1, 2 i 3, aby móc wykorzystać cały RAM i w razie potrzeby mieć dostęp do BASIC ROM-u. W tym celu należałoby zrezygnować z wymuszania na wejściach EXROM i GAME stałych stanów niskich, ale sterować tymi wejściami z jakiegoś układu dwustanowego, jak na przykład przerzutnik bistabilny. Pozostaje pytanie: jak spowodować, żeby przerzutnik zmienił stan? W tym celu można wykorzystać wyjścia I/O1 (pin 7) i I/O2 (pin 10). Wyglądałoby to tak:



Rys. 6

Wyjścia I/O1 oraz I/O2 pełnią podobną rolę jak ROML i ROMH, tyle że działają w innym obszarze adresowym. Zwykle mają one stan wysoki. Jeśli jednak procesor wykonuje jakąkolwiek operację (zapis, odczyt, przesuwanie bitów, itp.) na komórce pamięci o adresie z przedziału \$DE00-\$DEFF, wówczas I/O1 przyjmuje stan niski. Jeśli odwołuje się do adresów z przedziału \$DF00-\$DFFF, wówczas stan niski przyjmuje I/O2. W praktyce zostaje wygenerowany krótki impuls, który zupełnie wystarczy żeby „dać kopa” przerzutnikowi bistabilnemu i zmienić jego stan.

Wyjścia te pierwotnie miały służyć do zupełnie innych celów niż przełączanie pamięci; miały sterować komunikacją z urządzeniami zewnętrznymi, jako że firma CBM chciała C64 uczynić bardzo wszechstronnym. Planowano dołączać przez EXPANSION PORT stację dysków z równoległą szyną danych, a nawet procesor Z80. Ja wykorzystałem te wyjścia w specyficzny sposób do przełączania pamięci. Nie wiem, czy inni też tak robili, ale pewnie tak, bo nie ma tu żadnego wielkiego odkrycia i to się samo nasuwa na myśl. Układ przedstawiony na rys. 6 działa więc tak, że po włączeniu zasilania komputera lub po resecie przerzutnik bistabilny na wykorzystanym wyjściu ma stan niski, a więc konfiguracja pamięci jest taka jak na rys. 2, lub gdy zrealizujemy połączenie narysowane linią przerywaną – jak na rys. 3. Jest tak cały czas, dopóki program nie odwoła się do adresu z przedziału \$DF00-\$DFFF. Wówczas zostaje wygenerowany impuls na wyjściu I/O2 i przerzutnik osiąga na wyjściu stan wysoki, co powoduje przywrócenie konfiguracji pamięci z rys. 1. Komputer wtedy działa tak, jak gdyby w ogóle nie było kartridża. Jeśli jednak chcemy, żeby procesor ponownie „widział” CARTRIDGE ROM, wystarczy wykonać jaką-

kolwiek operację na dowolnym adresie z przedziału \$DE00-\$DEFF. Kartridż wtedy znowu jest podłączony do komputera. Przykład sekwencji rozkazów usuwającej z mapy pamięci CARTRIDGE ROM i przywracającej stan z Rys.1:

PHA

LDA \$DF00

PLA

Ponieważ do akumulatora wczytuje się wtedy jakaś „bzdurna liczba”, jego zawartość jest tymczasowo przechowywana w stosie. Oczywiście, żeby ponownie „włączyć” kartridż wystarczy tylko zmienić adres np. na \$DE00.

To, co przedstawiłem na rysunku jest tylko prostym przykładem. Jest tu wiele możliwych rozwiązań bardziej złożonych. Można np. użyć dwa przerzutniki bistabilne i sterować oddzielnymi wejściami EXROM i GAME, uzyskując w ten sposób wszystkie możliwe konfiguracje. Zamiast przerzutników można zastosować wpisywanie do rejestrów, co zwiększa możliwości oddziaływania na kartridż. Można używać również znacznie większe pamięci EPROM (64 KB i więcej) i podzielić je na przełączane 16 kilobajtowe banki pamięci. Można także umieścić w kartridżu dodatkowy RAM. Jak widać, przy projektowaniu kartridży do C-64 mamy całkiem sporo różnorodnych możliwości.

Na koniec jeszcze tylko wyjaśnienie, jak system operacyjny rozpoznaje, że kartridż w ogóle jest wetknięty do gniazda. Wymuszenie stanu niskiego na wejściach EXROM i GAME nie wywołuje żadnych zmian w stanie rejestrów komputera, więc system „nie wie”, że kartridż jest dołączony. Sprawę rozwiązano programowo. Przy każdym restarcie systemu, podprogram KERNALA sprawdza 5 bajtów pod adresami \$8004-\$8008 pod kątem obecności określonego hasła; są to liczby heksadecymalne C3, C2, CD, 38, 30, które po odjęciu od pierwszych 3 liczb wartości szesnastkowej 80 tworzą znaki CBM80 w kodzie ASCII. Gdy hasło zostanie wykryte, procesor wykonuje skok do ściśle określonego adresu w CARTRIDGE ROM i w ten sposób kartridż przełącza kontrolę nad systemem.

Dlatego też każdy projektant kartridży musi to hasło tam umieścić. Nawiasem mówiąc KERNAL testuje te adresy zawsze, nawet bez kartridża, badając wówczas RAM. Ten fakt wykorzystano w programach maszynowych z autostartem po resecie.

Romuald Drahokaupil

Przerabianie kartridży

Piszę ten artykuł po doświadczeniach z Black Box-ami V.8 i V.4. „Zdumpować” to raczej złe słowo, bo to by było tylko odczytanie zawartości i zapisanie tych danych – użyłbym może „zremasterować”. Tak więc remasterowanie carta należy zacząć od analizy jego działania – w jaki sposób on się zgłasza, czy jest widoczny czy niewidoczny, oraz sposobu bankowania, tzn. jak przełącza się banki pamięci carta. Kiedy już to wiemy, jest to dopiero połowa sukcesu. Później można, nie wnikając w to, co konkretnie robi program zaszyty w carcie, znaleźć wszystkie odwołania bankujące nim i pozmienić je w należyty do zakładanych celów sposób.

Pierwsze odkrycia koderów dotyczące sterowania kartridżami czytałem w jakimś bardzo starym polskim magu, chyba z początku lat 90-tych (jeśli nie końca 80-tych). Później Grabba opisał w C&A Actiona (Atraxowego). Istnieją dwa najpopularniejsze kartridże, w których można próbować podmieniać zawartość softu i mieć nadzieję, że wszystko zakończy się happy endem. Chodzi właśnie o Action i o Final III. Dlaczego? Otóż te dwa kartridże mają najpopularniejsze sposoby zgłaszania się i posiadają dodatkowe dla koderów ułatwienia. Jakże? O tym za moment.

Action i Final mają po 4 banki – Action po 8 KB, Final po 16 KB. Stąd wniosek, że Action ma 32 KB ROM, a Final 64 KB (!). Można to sprawdzić – rozbierając carta zobaczymy na EPROM-ie symbol 27c256 (w Actionie) i 27c512 (w Finalu). Rozmiar pamięci nie świadczy jeszcze o tym, w jaki sposób cart jest widoczny, bo przecież równie dobrze Final III mógłby mieć 8 banków po 8 KB (tak jak Ucart Suchego).

Dobrze. Rozważmy sobie, w jaki sposób zgłaszają się karty. Najczęstsze dwie możliwości to:
8 KB widoczne w obszarze \$8000 do \$9FFF
16 KB widoczne w obszarze \$8000 do \$BFFF

Pomiędzy tryb Ultimix i inne udziwnienia – te dwa sposoby załatwiają nam w zasadzie większość (jeśli nie wszystkie) naszych zachcianek, związanych z remasteringiem. To, co musimy zrobić, to ustalić jak zgłasza się wybrany cart. Ja to zrobiłem poprzez „macanie” I/O1 i I/O2 i porównywanie odczytywanych wartości z obszaru \$8000 do \$BFFF z zawartością RAM-u i BASIC-a. Jeżeli dane są różne, to wiemy, że w tym momencie „złapaliśmy” carta. Owo „macanie” to nic innego, jak wpisywa-

nie kolejnych wartości do wszystkich komórek obszaru I/O1 i I/O2, aż do skutku.

Obszar I/O1 to 256 adresów od \$DE00 do \$DFFF, a I/O2 o stronę dalej, czyli \$DF00 do \$DFFF obecny przy wartości #\$37 komórki \$01 – a więc LORAM, HIRAM i CHAREN = 1.

Black Box-y zgłaszały się w taki sposób:

BB3 8 KB od \$8000 do \$9FFF

(27c64 EPROM)

BB4 16 KB od \$8000 do \$BFFF

(27c128 EPROM)

BB8 16 KB od \$8000 do \$BFFF 2 banki

(razem 32KB 27c256 EPROM)

Co do BB8, to jeszcze ważne było, że po resecie widoczny był drugi bank (drugie 16 KB EPROM). BB4 i BB8 zgłaszały się podobnie, jak Final III, dlatego też zostały zremasterowane do niego.

Zajmijmy się teraz softwarowym bankowaniem cartów:

Action Replay bankował się poprzez zapis do komórki \$DE00 (I/O1)

Final III – zapis do \$DFFF (I/O2)

Istotne jest, że był to zapis konkretnych wartości, przez co można było uzyskać wybór banku kolejnych 8 czy 16 KB albo wyłączyć widoczność carta, np. w Actionie było to:

```
LDA #$00
STA $DE00
albo
LDA #$08
STA $DE00
```

W Black Box-ach, w zależności od wykonania i wersji, był to TYLKO zapis bądź ZAADRESOWANIE określonej komórki czy obszaru, np.:

```
STA $DF01
albo
STA $DF03
a czasami wystarczyło
CPY $DE34
```

Wszystko w zależności od wersji carta i elektroniki, przy czym istotne jest, że było to „latchowanie” po adresach, więc ważne było nie to, jaką wartość wpisujemy, tylko do jakiego adresu się odwołujemy, co w efekcie skracało kod – jak widać z 5 do 3 bajtów (zauważmy

tę różnicę!). Było to dla mnie niesłychanie niewygodne, ponieważ musiałem się gimnastykować, chcąc „wdusić” pięć bajtów w miejsce trzech!!

Dla przykładu podaję z głowy kod na przepisanie zawartości pierwszego banku BB8 z EPROM'u do RAM-u C64 pod ten sam adres (tzn. musimy mieć w expansion port oryginalny BB8):

```
* = $1000

SEI
LDA #$37
STA $01 ; LORAM HIRAM
CHAREN =1

LDY #$00
STY $FB
LDA #$80
STA $FC ; WEKTOR $FB WSKA-
ZUJE NA $8000

LDX #$40 ; $40 BLOKÓW
PO 256 BAJTÓW

STA $DF00 ; WŁĄCZAMY
WIDOCZNOŚĆ PIERWSZEGO
BANKU POD $8000-$BFFF
; JAK DRUGI BANK TO
ZMIENIĆ NA STA $DF04
; DLA BB4 CMP $DE01
LOOP

LDA ($FB),Y
STA ($FB),Y
INY
BNE LOOP

INC $FC ; NEXT BLOCK
INC $D020 ; BLINK
DEX
BNE LOOP

STA $DF03 ; WYŁĄCZAMY
WIDOCZNOŚĆ CARTA
; DLA BB4 CMP $DF00

CLI
RTS
```

Ważne jest, że robimy to z wyłączonymi przerwaniem, bo jeśli zgłosiłaby się w przerwaniu obsługa interpretera, to BASIC mamy „przesłonięty” kartridżem!!

Prosty i króciutki programik „demaskuje” nam zawartość carta i nie musimy wylutowywać EPROM'u, żeby odczytać jego zawartość programatorem :)

Uruchamiamy SYS 4096, dalej można pi-
sać swoje procedury, aby zgrać obszar RAM-u

\$8000-\$BFFF na dyskietkę albo chociaż podej-
rzeć, co „wróg” czyni.

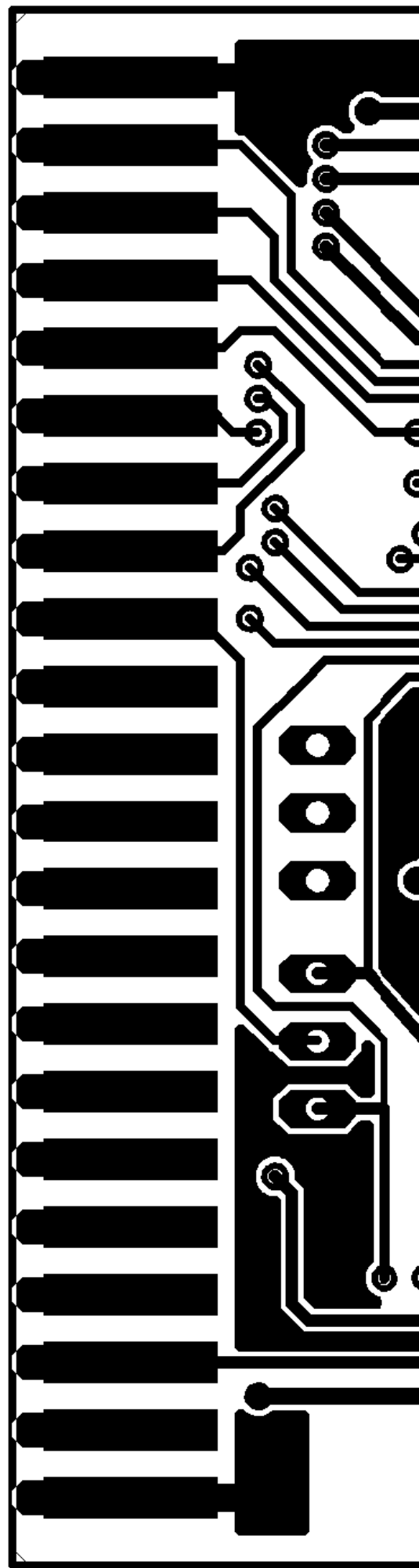
Dobrze więc. Skoro wiemy już, jak stero-
wać cartem, to możemy zabrać się do przeró-
bek związanych z softem, aby jego zawartość
zadziałała w innym hardware. Jak pisałem
wcześniej, nie wnikając w szczegóły, co dana
procedura robi, odnalazłem wszystkie od-
wołania w sofcie, odpowiedzialne za przełą-
czanie banków Black Boxa i pozmieniałem je
na odpowiednie dla Finala III. Niekomforto-
we było wyszukiwanie miejsca, aby zmienić
3 bajty na 5. Nieraz musiałem zmieniać także
adresy procedur je wywołujących, z uwagi na
pozmieniane offsety. Musiałem wyszukać, czy
przypadkiem zmieniana przeze mnie proce-
dura nie jest wywoływana z różnych miejsc
i zmieniać adresy JSR-ów. W dwóch przypad-
kach (banków) udało się zrobić to niemalże
bez pracy. Był to 1 bank BB8 i BB4. Chodzi
o kapitalną właściwość Finala III, który ma
włączoną widoczność pewnego obszaru obec-
nie używanego banku pod \$DF00 do \$DFFF.
W C64 jest to obszar I/O, gdzie normalnie nikt
się nie odwołuje, nie umieszcza tam swoich
programów, tak samo jak na adresach VIC'a
czy SID'a. Więc w obszarze tym nawet po wy-
łączeniu widoczności carta pod \$8000 nadal
jest widoczna zawartość tych danych! Chętnie
z tego skorzystałem i udało mi się zrobić taką
sztuczkę, że akurat jest to tekst winiety HEAD
FIT i w miejscu spacji wpisałem krótkie proc-
ki bankujące kartridżem, a wszystkie w tym
banku odnalezione odwołania bankujące kar-
tą zmieniłem na JSR-y w to miejsce widoczne
pod \$DF7C i \$DF81. Pozostało mi tylko zmie-
nić procedurę uruchamiającą HEAD FIT, tzn.
program HEAD FIT jest uruchamiany w taki
sposób, że jest przepisywany z ROM-u carta do
RAM-u, następnie wyłączana jest widoczność
carta i uruchamiany jest HEAD FIT. Ja więc
dodałem jeszcze kilka bajtów, które w RAM-
ie nadpisywały moją procedurkę spacjami, tak
jak było w oryginale, dzięki czemu przeróbka
BB4 odbyła się dosłownie w kilka godzin.

BB4 i BB8 ruszyły u mnie na 1541U jako
custom FC3 i dają 99,9% szans, że kto wymieni
EPROM w Final III na dane zawarte w binie
udostępnionym na CSDB, to będzie mógł się
cieszyć zamianą zwykłego Finala w nasz polski
produkt ! :))))))))))

Cóż, mam nadzieję, że naświetliłem ideę
do dalszych samodzielnych doświadczeń
i działań. Po artykule o karcie Suchego i wy-
wiadzie z twórcą Black Box-ów można uzyskać
trochę wiedzy o tych tajemniczych „czarnych
pudełkach” ;-)

I to by chyba było na tyle...

Wegi



Rozszerzenie + 60K

Kiedy zabieramy się za tworzenie programów, w pewnym momencie uświadamiamy sobie, że 64 KB to zbyt mało pamięci na nasz program i aby pisać na całą pamięć Commodore 64, potrzebna będzie nam dodatkowa pamięć. Jednak zanim napiszę jak sobie z tym radzić, to przedstawię metodę tworzenia różnych produkcji przez koderów. W latach 80-tych większość z nich tworzyła w monitorze języka maszynowego, w którym operuje się na pamięci bezpośrednio. W monitorze nie zgrywa się „źródłówek” programu, ale odpowiedni fragment pamięci. Taka praca była bardzo męcząca i zajmowała wiele czasu. Wpisywanie poleceń odbywało się bezpośrednio i nie można było łatwo ingerować w kod programu. Jeśli zapomnieliśmy napisać jakiejś procedury w kodzie, trzeba było pisać od początku albo dać odwołanie do danej procedury, która leżała gdzieś indziej w pamięci. Takie rozwiązanie nie zawsze było dobre. Zazwyczaj w kodzie panował duży bałagan, a najważniejsza sprawa to fakt, że kodu programu nie można przenieść bez poprawek pod dowolny adres pamięci. Dopiero pojawienie się w latach 90-tych programu Turbo Assembler zrewolucjonizowało tworzenie oprogramowania na C64. Jak na tamte czasy był to bardzo ciekawy i udany program użytkowy, pomocny w programowaniu i znacznie ułatwiający pracę koderowi. Powstało wiele jego modyfikacji, bo każdy koder znajdował w nim coś do usprawnienia.

Niestety każdy, nawet rewelacyjny program okazuje się mało przydatny, kiedy brakuje nam pamięci na tworzony przez nas soft. Załóżmy, że piszemy program, który będzie zajmować całą pamięć komputera. Kod tego programu zajmuje 30 KB, a resztą są dane (tj. grafika, muzyka oraz teksty). Nasuwa się pytanie: a gdzie pamięć na edytor do pisania tego programu? Za każdym razem, kiedy chcemy sprawdzić wyniki naszej pracy, musimy zgrywać źródło programu i po jego uruchomieniu musimy na nowo uruchamiać edytor i ponownie wgrywać źródło. Po pewnym czasie staje się to bardzo

uciążliwe i zwyczajnie zaczyna wkurzać. Jednak znalazły się różne pomocne rozwiązania, ułatwiające pracę koderowi z komputerem. Tymi rozwiązaniami okazały się rozszerzenia pamięci. Najbardziej znane to firmowe rozszerzenie firmy Commodore – RAM Expansion Unit (znane bardziej jako REU) o pojemności od 128 KB do 2 MB. Niestety, to rozszerzenie było mało popularne w Polsce, a jak się już pojawiło, to było bardzo drogie. Dostępne było także polskie rozszerzenie o nazwie Ram-Cart o pojemności 64 KB lub 128 KB podtrzymywane baterijnie. Jednak to urządzenie miało jedną dużą wadę – było wkładane do portu kartridża, przez co nie mogliśmy użyć naszego ulubionego karta. Inną wadą takich rozszerzeń był brak oprogramowania, które umożliwiałoby ich lepszą współpracę ze stacją dysków. Mało kto w Polsce myślał o rozszerzeniu montowanym w środku komputera. Jego wadą jest to, że trzeba znać się na elektronice, przez co wiele osób nie mogło takiego rozszerzenia zamontować samodzielnie. Pierwszą osobą, która stworzyła proste w budowie rozszerzenie i napisała do niego oprogramowanie był Adam Buława, znany bardziej jako Mr. Fiz z grupy Samar. Opublikowanie schematu, „przerobienie” Turbo Assemblera pod to rozszerzenie oraz napisanie programu kopiującego było strzałem w dziesiątkę, który przekonał ludzi (a szczególnie koderów i swaperów) do montowania tego rozwiązania w swoim komputerze. Na początku powstały dwie wersje rozszerzenia. Jedna przystosowana była do starej płyty C64, a druga nowej, którą w tamtych czasach miała większość użytkowników tego komputera. Pomysł zrobienia czegoś takiego powstał, jak to często bywa, z potrzeby ułatwienia sobie pracy. Kiedy autor rozszerzenia pochwalił się nim innym i pokazał jak działa, zaraz zaczęto je montować. Rozszerzenie to stało się bardzo popularne w Polsce i w czasie świetności polskiej demosceny posiadało je ponad 100 osób (zresztą mogła być ich znacznie większa liczba). Pierwszą osobą, która zrobiła sobie takie rozszerzenie poza

graniami Polski był węgierski koder i elektronik Soci. On też usprawnił schemat rozszerzenia i rozbudował je do 256 KB (choć powstały – na specjalne zamówienie – rozszerzenia 256 KB autorstwa Mr. Fiz’a, które jednak posiadało bardzo mało osób).

Jeżeli chodzi o oprogramowanie na +60K, to nie jest tego dużo. Podstawa to przerobiony Turbo Assembler z dodanymi rozkazami niepublikowanymi, a prócz tego kilka programów kopiujących, ram-dyski, testery rozszerzenia, jakieś sample oraz kilka programów przerobionych tak, aby działały z tym rozszerzeniem. Wśród produkcji demosceny są tylko trzy, które wykorzystują ten sprzęt. Warto wspomnieć jeszcze, że istnieje emulacja tego rozszerzenia w emulatorze VICE.

Zalety:

- dobre rozwiązanie dla koderów, którzy tworzą duże produkcje (gry, dema, ...),
- szybsze kopiowanie dyskietki, szczególnie jak ma się jeszcze zrobionego bursta,
- montowane w środku komputera, przez co ma się dostęp do portu expansion.

Wady:

- mała ilość dostępnej pamięci, szczególnie jak ktoś chce skopiować dyskietkę na raz,
- VIC nie widzi rozszerzenia, więc kombinacje z grafiką odpadają,
- kolizja z programami, które wykorzystują adres \$d100,
- obecnie projekt rozszerzenia jest lekko przestarzały, bo powstało wiele nowych i lepszych rozwiązań
- wymaga znajomości elektroniki, aby móc sobie je zrobić.

Instalujemy +60K

Zanim przejdę do meritum sprawy pragnę poinformować, że wykonanie rozszerzenia RAM wymaga znajomości podstaw elektroniki - inaczej może to grozić uszkodzeniem sprzętu lub ewentualnym uszkodzeniem samego siebie :P Opiszę tu „krok po kroku” sposób, w jaki ja je wykonałem (oczywiście nie trzeba brać wszystkiego pod uwagę).

Narzędzia

Do wykonania projektu użyłem następujących narzędzi: lutownica (chyba każdy z nas miał z nią do czynienia), odsysacz (przydatne urządzenie, szczególnie przy wyciąganiu scalaków, na Allegro można kupić za parę złotych), cyna, kalafonia, szczypce, laminat (do wykonania płytki drukowanej, koszt około 3 zł), wytrawiacz (do wytrawiania płytki drukowanej, koszt też około 3 zł), pisak do płytek drukowanych, mini wiertarka + wiertło, kilka kabelków.

Schemat

Przeglądamy się schematowi i kompletujemy potrzebne do jego wykonania części (Rys. 1). Jak widać są to:

- US1 i US2 41464 – chciałem kupić, lecz koszt (100 zł) mnie przeraził, postanowiłem więc

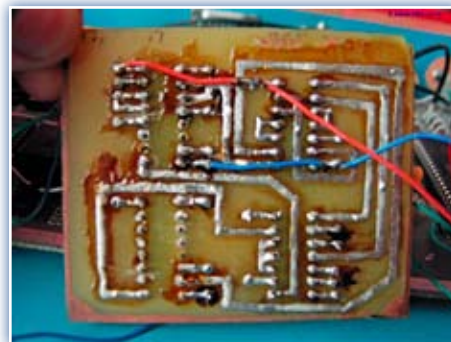
poświęcić drugą komodę. I w tym momencie przyda nam się odsysacz. Jak mamy już drugi C64, to delikatnie grzejemy nóżki, odciągamy cynę i wyciągamy kości. Pamiętajmy, aby ich nie przegrzać. Jeżeli mamy małą wprawę, zawsze można potrenować na jakichś starych płytkach.

- US3 74LS02 – ja zastosowałem szybszy 74HCT02, bo tamtego nie dostałem (też działa).
- US4 74LS00
- US5 74LS139
- US6 74LS174
- włącznik on/off
- podstawki pod układy 2x dil 14, 2x dil 16, 2x dil 18

Płytką drukowaną

Zrobienie płytki nie jest konieczne, ponieważ możemy to zrobić „na pająka”, ale wygląd jest tragiczny. Dlatego żeby zredukować liczbę kabli, poświęcamy swój czas i robimy płytkę. Ja rozrysowałem to po swojemu i wygląda to tak jak na Rys. 2. Teraz musimy to przenieść na laminat, wcześniej czyścimy go drobnoziarnistym papierem ściernym i odtłuszczamy. Rozrysowujemy układ ścieżek za pomocą markera do płytek drukowanych na wyczyszczonym laminacie, przycinamy na odpowiedni wymiar

i wiercimy otwory na nasze układy. Gdy wszystko pasuje, przystępujemy do wytrawiania płytki. Ja kupiłem wytrawiacz, który rozpuściłem w ciepłej wodzie (instrukcja na opakowaniu), wrzucamy naszą płytkę i czekamy. Miejsca niepomalowane markerem znikną. Po wyczyszczeniu gotowej płytki наносимy na ścieżki cynę z kalafonią i sprawdzamy, czy nie ma jakichś niepotrzebnych połączeń, żeby nie było zwarcia.



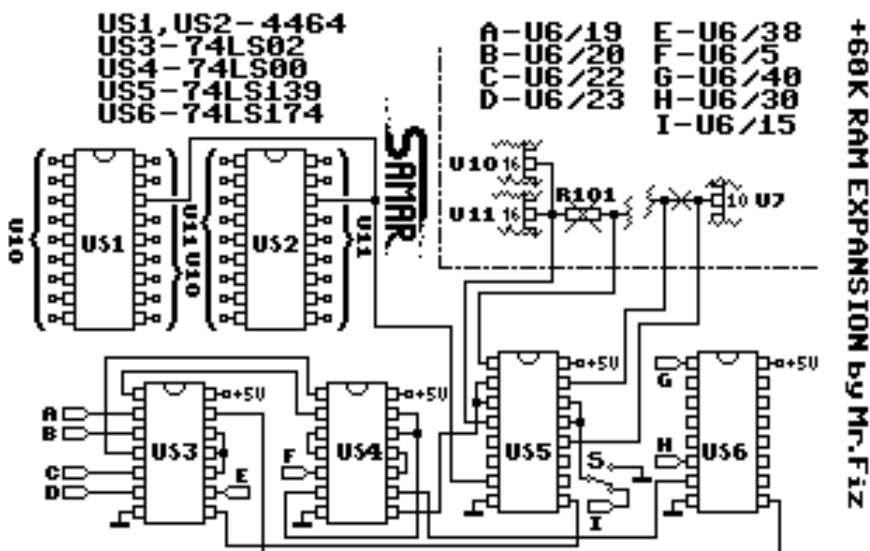
Montaż

Lutujemy do naszej płytki cztery podstawki pod układy US3, US4, US5, US6. Układy US1 i US2 montujemy w naszym Commodore i tutaj ja wyciągnąłem układy i wlutowałem podstawki foto5. Następnie zlutowałem je jeden na drugim, pozostawiając nóżki numer 16, które łączymy ze sobą kablem i podłączamy do naszej płytki do nóżki numer 7 US5. Teraz podłączamy US3 na naszej płytce z układem U6 na płycie głównej (Foto 1), czyli lutujemy kable kolejno od układu US3 do U6:

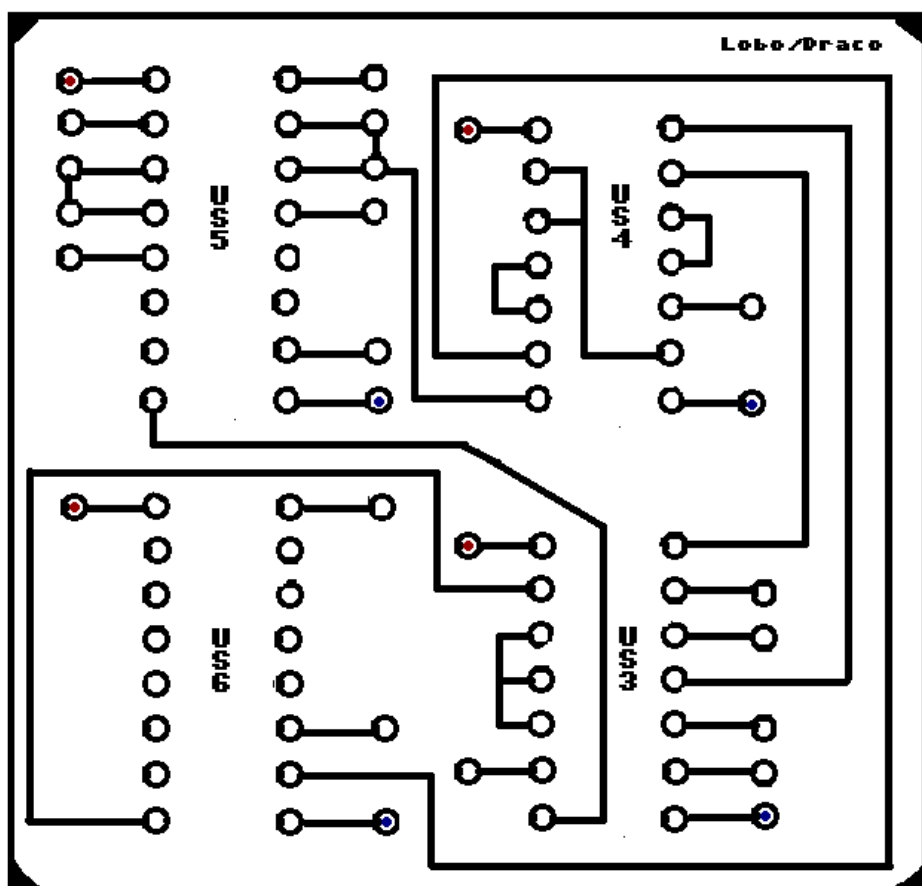
	US3	U6
Nóżka :	2	- 19
	3	- 20
	5	- 22
	6	- 23
	9	- 38

Ja lutowałem od spodu płyty do układu U6, pamiętajmy w tej sytuacji o kolejności nóżek patrząc zawsze od góry. Robimy to samo z układem US4 i US6 czyli:

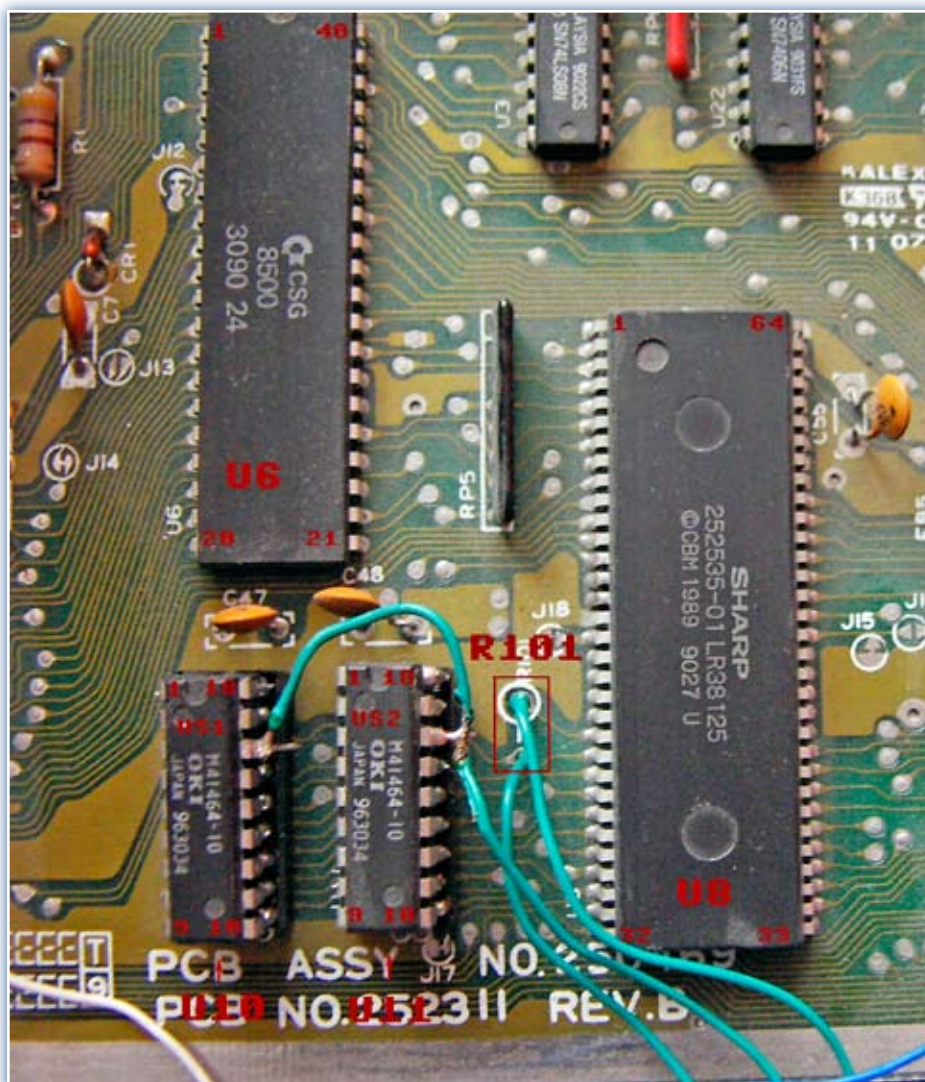
	US4	U6
Nóżka :	5	- 5
	US6	U6
Nóżka :	1	- 40
	6	- 30



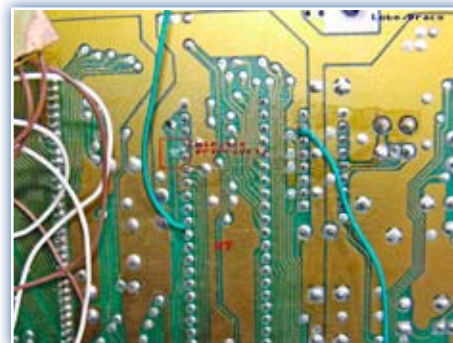
Rys. 1



Rys. 2



Do układu US5, jak widać na schemacie, podłączamy włącznik I/O dokładnie do nóżek numer 13 i 14, następnie do nóżki układu U6 numer 15, a trzecią nóżkę włącznika do masy. Na płycie odnajdujemy rezystor R101, który wylutujemy. W jego miejsce lutujemy dwa kabełki. Jeden podłączamy do nóżki numer 4 US5 – jest to ten, który będzie teraz łączył nóżki numer 16 układu U10 i U11 na płycie, czyli nasz RAM. Drugi kabełek podłączamy do nóżki numer 1 US5. Szukamy teraz układu U7 na płycie i przerywamy ścieżkę łączącą nóżkę numer 10 z nóżką numer 40 U8 (ja to zrobiłem od spodu).



Teraz łączymy US5 nóżkę 12 z układem U7 nóżka 10 i US5 nóżkę 15 z U8 nóżka 40. Do masy podpinamy US3 i US4, będą to nóżki numer 7 a do US5 i US6 numer 8. Zasilanie +5V podpinamy do US3 i US4 nóżek numer 14, a do US5 i US6 numer 16 (zasilanie +5V podłączyłem tutaj)



Test

Gdy upewnimy się, że wszystko jest połączone zgodnie ze schematem, przejdźmy do testu. Pamiętajmy, aby dwa razy sprawdzić nasze wykonanie. Lepiej mieć 100% pewności, chociaż nigdy jej nie ma.

Podłączamy zasilanie i uruchamiamy Commodore, załączamy nasze +60K. Gdy wszystko wygląda na OK, wczytujemy program, np. +60K Test Memory by TSD (<http://noname.c64.org/csdb/release/?id=14441>) uruchamiamy, klikamy literę T i sprawdzamy, czy wszystko jest w porządku.

Miłego majsterkowania i powodzenia.

+32KB RAM Expansion dla stacji dysków 1541 II

Z tej strony wita was Klax. Chciałbym zaprezentować wam projekt rozszerzenia pamięci +32KB, który dedykowany jest do stacji dysków 1541 II. Rozszerzenie funkcjonuje w zakresie \$4000-\$bfff i potrzeba zaledwie kilku elektronicznych elementów, by je wykonać:

- jeden scalak 74HCT139 (dekoder)
- 2 diody 1N4148
- jeden rezystor 10 KOhm
- pamięć 62256 lub kompatybilna, najlepiej wąska (dostępna np. na bardzo starych pecetowskich płytach głównych)
- odrobina dobrej intencji i trochę wolnego czasu.

Teraz przejdę do wyjaśnienia, w jaki sposób zamontowałem rozszerzenie pamięci w mojej stacji dysków. Do tego celu nie projektowałem żadnej płytki, ponieważ cały projekt nie jest zbyt skomplikowany. Pierwszy krok: wyjmij ROM z płyty głównej stacji dysków, odegnij 22 pin i wsadź z powrotem do podstawki. W następnej kolejności spróbuj odnaleźć scalaka 74LS42, a następnie ścieżkę biegnącą pomiędzy 4 i 5 pinem. Ta ścieżka jest bardzo charakterystyczna, ponieważ pierwotnie poprowadzona jest na drugiej stronie płyty głównej, a następnie przez przelotkę przechodzi na górną. Najłatwiej zlokalizować ją pomiędzy 4 a 5 pinem kości 74LS42 i w tym miejscu należy ją przeciąć.



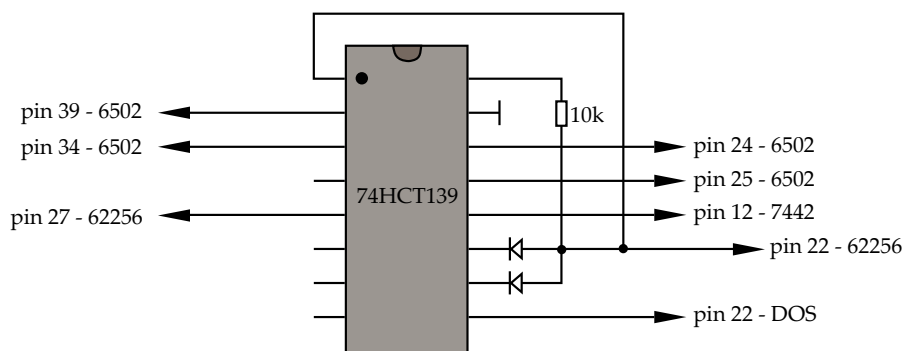
Mając teraz do wyboru dwie drogi: dołączenie dekodera lub podłączenie RAM'u, proponuję wybrać najpierw tę drugą opcję. Należy zgąć wszystkie piny kości RAM'u 62256 pod kątem 90 stopni na zewnątrz od ich pierwotnego położenia. Tę czynność należy wykonywać ostrożnie, albowiem istnieje możliwość ułamania jednego, bądź więcej pinów, zwłaszcza gdy kość pochodzi ze starej, pecetowskiej płyty głównej. Po udanym zabiegu z pinami trzeba odwrócić płytę główną stacji i przylutować kość RAM'u do nóżek kości ROM'u (dołączone do artykułu zdjęcie pozwoli zrozumieć ideę tego połączenia). Podczas lutowania warto pamiętać, aby piny RAM'u o numerach 1, 22 i 27 nie zostały przylutowane.

Kolejnym krokiem w montażu rozszerzenia jest instalacja dekodera (74HCT139). Połączeń należy wykonać zgodnie ze schematem, nie zapominając o diodach i rezystorze. Można ponownie posłużyć się metodą wyginania nóżek scalaka. Nie można także zapomnieć o podłączeniu odpowiedniego kabelka do pinu 22 kości ROM'u.

Kiedy lutowanie dobiegnie końca, a włączona do sieci stacja dysków zachowuje się prawidłowo, oznacza to, że wykonanie rozszerzenia pamięci prawdopodobnie zakończyło się sukcesem. Wówczas należy przetestować pamięć przy pomocy monitora dostępnego w Action Replay'u. Należy wejść do niego i wykonać polecenie @*8 a następnie m4000 i nadpisać jakąś wartość komórki, zatwierdzając czynność klawiszem Return. Potem ponownie wykonać polecenie m4000 i sprawdzić, czy wpisana przed chwilą wartość pozostała na swoim miejscu. Jeżeli tak, to oznacza, że nasze rozszerzenie pamięci działa prawidłowo!

*Autor rozszerzenia, tekstu i zdjęć: Klax
Edycja artykułu i poprawki: V-12/Tropyx*

W nieco skróconej wersji tekst ukazał się pierwotnie w anglojęzycznym magazynie dyskowym Vandalism News #44/Oslaught/Wrath Designs z 2005 roku.



pin 1 połączyć z pinem 24 - 6502

Programować

w asemblerze każdy może

obroty 2D oraz rysowanie pikseli

Pewnego dnia znalazłem ciekawy efekt w sieci (kaleidoscope drawing) i zacząłem się zastanawiać, czy da się go przenieść na C64. Oczywiście było to możliwe. Natrafiłem na szereg problemów, jednak rozwiązywanie ich była fajną rozrywką.

Dużo osób zniechęca się do programowania przez obszerne kursy, tymczasem nie jest to takie trudne. Każdy może spróbować. Dziś pokażę, jak wykonać obrót punktu oraz narysować go w asemblerze. Postaram się opisać wszystko jak najbardziej zrozumiale.

Na początku zorientowałem się, że do efektu potrzebne będą obroty w przestrzeni dwuwymiarowej (czyli na płaszczyźnie) wokół środka ekranu.

Z geometrii wiem, że obrót zapisuje się następującym wzorem:

$$\begin{aligned}x &= x0 * \cos(a) + y0 * \sin(a) \\ y &= y0 * \cos(a) - x0 * \sin(a)\end{aligned}$$

Zapis algorytmu obrotów punktu wokół środka ekranu w Basicu wyglądałby tak:

zmienne występujące w programie:
XX – współrzędna X punktu wejściowego
YY – współrzędna Y punktu wejściowego

SZER – zawiera szerokość ekranu w pikselach bieżącego trybu graficznego

WYS – zawiera wysokość ekranu

ILE – ilość punktów powstających za pomocą obrotów – w następujący sposób weźmy wartość 12: początkowy punkt obracamy o 30 stopni, w kolejnym obrocie początkowe współrzędne obracamy o 60, później 90 itd. Ostatni punkt wypadnie w tym samym miejscu co pierwszy, bo obrócimy się o pełne 360 stopni.

```
...
100 ILE=12
110 FOR I=0 TO ILE
115 REM *TRANSLACJA*
120 X0 = XX-SZER/2
130 Y0 = YY-WYS/2
135 REM *KATY MUSIMY ZAMIE-
    NIC NA RADIANY*
160 RAD = I*(360/ILE) * π/180
140 X = X0*COS(RAD) + Y0*SIN(RAD)
150 Y = Y0*COS(RAD) + X0*SIN(RAD)
155 REM *KOLEJNA TRANSLACJA*
160 X = X+SZER/2
170 Y = Y+WYS/2
180 GOSUB 9000
190 NEXT I
9000 REM *PODPROGRAM
    WYŚWIETLAJĄCY PIKSELE O
    WPÓLRZĘDNYCH (X,Y)*
...
9900 RETURN
```

Pojawia się tu szereg problemów.

Po pierwsze: w asemblerze nie da się używać w prosty sposób ułamków – sin i cos daje liczby zmiennoprzecinkowe od -1 do 1. Możemy tu skorzystać ze sztuczki genialnej w swojej prostocie – z przemienności mnożenia.

Weźmy prosty przykład: chcemy wykonać działanie $X * 0,01$. W tym celu:

- najpierw mnożymy ułamek przez liczbę całkowitą (najlepiej potęgę 2) i odrzucamy część ułamkową: $0,01 * 64 = 6,4 \approx 6$

- następnie tak otrzymaną liczbę (zamiast liczby zmiennoprzecinkowej) mnożymy przez X, otrzymujemy: $X * 6$

- na końcu wynik dzielimy przez tą samą liczbę, przez którą mnożyliśmy ułamek na początku. Ostateczne rozwiązanie będzie równe $X * 6 / 64$.

Co nam to daje? Wystarczy, że w tablicy będziemy przechowywać wymnożone przez 64 wartości sin, cos, odczytywać wartość, mnożyć przez liczbę, a na końcu dzielić przez 64.

Kolejnym problemem jest sposób, w jaki asembler przechowuje wartości ujemne. Każda komórka ma 8 bitów, czyli może przechowywać wartości od -127 do 128. Za znak odpowiada wtedy najstarszy bit liczby, a cała liczba jest zapisana w kodzie U2. Jeżeli liczba ma wartość dodatnią, to nic nie zmieniamy, wszystkie liczby do 127 możemy normalnie zapisywać jak w zwykłym kodzie binarnym. Liczby ujemne zapisuje się biorąc wartość bezwzględna, zmieniając wszystkie jedynki na zera, a zera na jedynki (wykonując negację bajtu), a następnie dodając 1. Oto jak zamienić liczbę dodatnią na liczbę ujemną:

- ładujemy określony bajt z pamięci (tutaj dodatni o wartości od 0 do 127)

LDA bajt

- negujemy wszystkie bity - w asemblerze C64 nie ma polecenia negacji, ale wykonujemy je przez polecenie EOR (czyli Exclusive OR) z maską 255:

EOR #\$ff

- na końcu dodajemy liczbę jeden. Przed dodaniem trzeba wyzerować znacznik przeniesienia

CLC
ADC #\$1

I już w akumulatorze mamy liczbę ze znakiem -. Aby pozbyć się tego znaku, należy postąpić identycznie.

Kolejną rzeczą jest rysowanie pikseli. Niestety nie znajdziemy jej w Basicu, ale będziemy musieli ją napisać od podstaw w asemblerze.

Najszybszy sposób jaki znam, polega na ustawieniu znaków kolumnami w następujący sposób:

```
@O...
AP
BR
CS
DT
...
O
```

Następnie można korzystając z odpowiednich tablic rysować piksel w wybranym miejscu na ekranie (definiując kształty znaków) za pomocą kilku instrukcji. Wadą tego sposobu jest to, że możemy korzystać wtedy tylko z niewielkiego wycinku ekranu (127x127piksli). My zajmujemy się rysowaniem w HiRes, ponieważ będzie lepszy do tego typu efektu – mamy niewiele pikseli do narysowania i wygenerowany obraz przykryje większą część ekranu. Będziemy do tego potrzebowali specjalnie przygotowanych tabel, zawierających poszczególne linie i kolumny. Następnie na podstawie tych tabel będziemy mogli szybko wyznaczyć, do jakiej komórki pamięci wpisywać dane, aby piksel pokazał się na ekranie. Poprawiony program w Basicu:

```
...
100 ILE=12
105 REM*LICZYMY TABLICE SINU-
    SÓW*
110 ALPHA=(360/ILE) * π/180
120 FOR I=0 TO ILE
130 COS_TAB(I) = COS(ALPHA)
135
140 SIN_TAB(I) = SIN(ALPHA)
150 ALPHA=ALPHA+(360/ILE) *
    π/180
160 NEXT I
200 REM *TERAZ W COS_TAB
    MAMY WARTOŚCI KOLEJNO
    DLA 30, 60, 90 itd. STOPNI
210 FOR I=0 TO ILE
215 REM *TRANSLACJA*
220 X0 = XX-SZER/2
230 Y0 = YY-WYS/2
240 REM *LICZYMY OBROTY UŻY-
    WAJĄC TABEL*
250 X = X0*COS_TAB(I)/64 +
    Y0*SIN_TAB(I)/64
260 Y = Y0*COS_TAB(I)/64 +
    X0*SIN_TAB(I)/64
275 REM *KOLEJNA TRANSLACJA*
280 X = X+SZER/2
290 Y = Y+WYS/2
300 GOSUB 9000
310 NEXT I
9000 REM *PODPROGRAM WY-
    ŚWIETLAJĄCY PIKSELE O
    WPÓŁRZĘDNYCH (X,Y)*
...
9900 RETURN
```

Poniżej znajduje się cały listing programu – jest to po prostu przeniesienie tego programu z Basicu do assemblera, uzupełnione dodatkowo o funkcję rysującą piksel i umożliwiającą sterowanie. Tabeli nie będziemy już generować, umieścimy ją bezpośrednio w programie:

```
*= $0801

x0 = $f1
y0 = $f3
mnozna = $03
mnoznik = $04
sign = $05
storeplot = $d3
x = $f6
y = $f7
xx = $f8
yy = $f9
tmp = $fa
i = $fb

screen = $2000

lda #0
sta storeplot
sta storeplot+1
```

Obliczanie tabel potrzebnych do rysowania pikseli na ekranie w trybie HiRes:

```
settbadr
    ldx #$00
    lda #>screen
    stx $fb
    sta $fc
settb2
    lda $fb
    sta tbadlo,x
    lda $fc
    sta tbadhi,x

    lda $fb
    clc
    adc #$40
    sta $fb

    lda $fc
    adc #$01
    sta $fc
    inx
    cpx #25
    bcc settb2
```

Inicjalizacja grafiki, ustawienie trybu i miejsca bitmapy itd.

```
initgraph
;lines 5 - 90 from tomaa
basic

; lda #$06
; sta 53280
```

```
lda 53272
ora #$08
sta 53272
```

```
lda 53265
ora #32
sta 53265
```

Wypełnienie pamięci kolorów bajtem \$10, zawierającym informacje o kolorze tła i kolorze rysunku (w tym przypadku rysunek będzie w kolorze białym – kod \$1, a tło czarne –kod \$0).

```
ldx #$00
lda #$10
fill1
    sta $0400,x
    sta $0500,x
    sta $0600,x
    sta $06f8,x
    inx
    bne fill1

    ldx #$20
    stx $fc
    ldy #$00
    sty $fb
    lda #$00
```

Wypełnienie pamięci obrazka bajtem \$0

```
fill2 sta ($fb),y
    iny
    bne fill2
    inc $fc
    dex
    bne fill2

    lda #20
    sta xx
    sta yy

start
```

Testowanie, jakie bity w rejestrze odczytującym stan joysticka port III \$dc00 są zapalone i sterowanie za pomocą tego głównym punktem, którego używamy do obracania.

```
lda #%00000001
bit $dc00
bne dol
dec yy

dol lda #%00000010
bit $dc00
bne gora
inc yy

gora lda #%00000100
bit $dc00
bne lewo
dec xx
```



```

lewo  lda #%00001000

      bit $dc00
      bne fire
      inc xx

fire  lda #%00010000
      bit $dc00
      bne start

      lda xx
      sta x0
      lda yy
      sta y0
      jsr pix

      ldy #0 ;for(i=1;i<12;i++)
      sty i

```

W pętli obracamy jeden punkt o kolejne wielokrotności 30 stopni, ponieważ gdybyśmy próbowali obracać kolejne punkty o 30 stopni, wówczas nastąpiłaby stopniowa kumulacja błędu (tabelki sin i cos zawierają jedynie wartości przybliżone)

```

loop

```

Przesunięcie początku układu współrzędnych na środek ekranu – trzeba odjąć od współrzędnych wejściowych xx ½ szerokości, a od yy ½ wysokości, następnie po obliczeniach obrotu dodać te wartości do wyniku

```

lda xx
sec
sbc #$60
sta x

lda yy
sec
sbc #$60
sta y

lda x
sta mnozna
ldy i
lda cos,y ;cos 30*i
sta mnoznik
jsr mul ;mnożymy wynik jest od
razu dzielony na 64
;-----
sta tmp
lda y
sta mnozna
ldy i
lda sin,y
sta mnoznik
jsr mul
;-----
clc
adc tmp ;tutaj mamy wynik
x = x0 * cos(a) + y0 * sin(a)

```

```

clc
adc #$60 ;dodajemy ½ szerokości
sta x0

lda x
sta mnozna
ldy i
lda sin,y
sta mnoznik
jsr mul
;-----
sta tmp
lda y
sta mnozna
ldy i
lda cos,y
sta mno.nik
jsr mul
;-----
sec
sbc tmp; a tu y = y0 * cos(a) - x0
          * sin(a)

clc
adc #$60 ;dodajemy ½ wysokości
sta y0

jsr pix ; rysowanie piksela
          o współrzędnych (x0, y0)

ldy i
iny
sty i

cpy #11
bne loop

jmp start

```

Mnożenie z artykułu Wegiego

```

mul
lda mnozna
ldx mnoznik
eor mnoznik ;check sign of
sta sign ;result
bpl sign2
lda mnozna
beq sign2 ;result will be 0 not < 0
lda mnoznik
beq sign2
lda #"-"
bne sign3
sign2  lda #$20
sign3  sta sign
lda mnozna
bpl sign4
eor #$ff
clc
adc #$01
sta mnozna
sign4  lda mnoznik
bpl sign5

```

```

eor #$ff
clc
adc #$01
sta mnoznik
sign5  lda mnozna
ldx mnoznik
;-----
unsmult
sta mnozna
stx mnoznik
lda #$00
ldx #$08
mnoze  ror mnoznik
bcc skok
clc
adc mnozna
skok   ror a
dex
bne mnoze
ror mnoznik
ldx mnoznik

```

Dzielenie dwubajtowe przez 64 - po przesunięciu w prawo w przeniesieniu jest 0 lub 1, w zależności od tego, jaki był najmłodszy bit (pierwszy z prawej) ror pobiera bit z przeniesienia i wstawia w najstarszej pozycji (bit pierwszy z lewej), następnie całość przesuwamy w prawo. Przesunięcie w prawo odpowiada dzieleniu przez 2, postępujemy tak 6 razy, bo $2^6 = 64$

```

lsr a
ror mnoznik
lsr a
ror mnoznik
lsr a
ror mnoznik
lsr a
ror mnoznik
lsr a
ror mnoznik
lsr a
ror mnoznik
lsr a
ror mnoznik

```

W zmiennej sign jest zachowany znak operacji, w razie potrzeby dodajemy znak minus w kodzie uzupełnień do 2.

```

lda sign
cmp #"-"
bne unsign
lda mnoznik
eor #$ff
clc
adc #$01
rts
unsign
lda mnoznik
rts

```

Procedurka rysująca piksele

```

pix
    lda x0
    cmp #200-8
    bcc dj1
    rts

dj1
    lda y0
    cmp #200
    bcc dj2
    rts

dj2
    lda x0
    clc
    adc #64
    sta x0

    lda y0
    lsr a
    lsr a
    lsr a
    tax

    lda y0
    and #$07
    tay
    lda x0
    and #$f8
    clc
    adc tbadlo,x
    sta storeplot

    lda tbadhi,x
    adc x0+1
    sta storeplot+1

    lda x0
    and #$07
    tax

    lda (storeplot),y
    ora tbbit,x
    sta (storeplot),y

    rts
    
```

Tabelki potrzebne przy rysowaniu pikseli

```

tbbit
    .byte %10000000
    .byte %01000000
    .byte %00100000
    .byte %00010000
    .byte %00001000
    .byte %00000100
    .byte %00000010
    .byte %00000001
    
```

tbadlo

```

    .byte 0,0,0,0,0
    .byte 0,0,0,0,0
    .byte 0,0,0,0,0
    .byte 0,0,0,0,0
    .byte 0,0,0,0,0
    
```

tbadhi

```

    .byte 0,0,0,0,0
    .byte 0,0,0,0,0
    .byte 0,0,0,0,0
    .byte 0,0,0,0,0
    .byte 0,0,0,0,0
    
```

Tabelki sin i cos dla wartości kątów:
30, 60, 90, 120, itd...

cos

```

    .byte $37,$20,$00,$e0,$ca,$c1
    .byte $ca,$e1,$00,$20,$37,$40
    
```

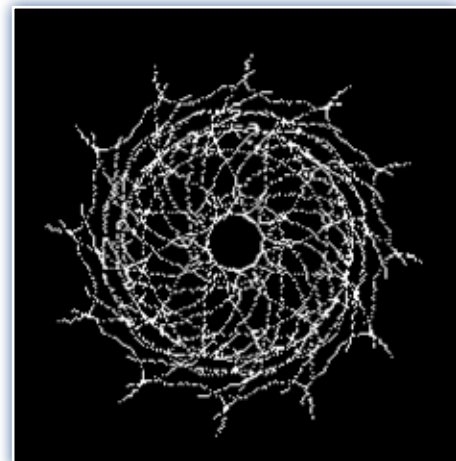
sin

```

    .byte $20,$37,$40,$37,$20,$00
    .byte $.1,$ca,$c1,$ca,$e1,$fb
    
```

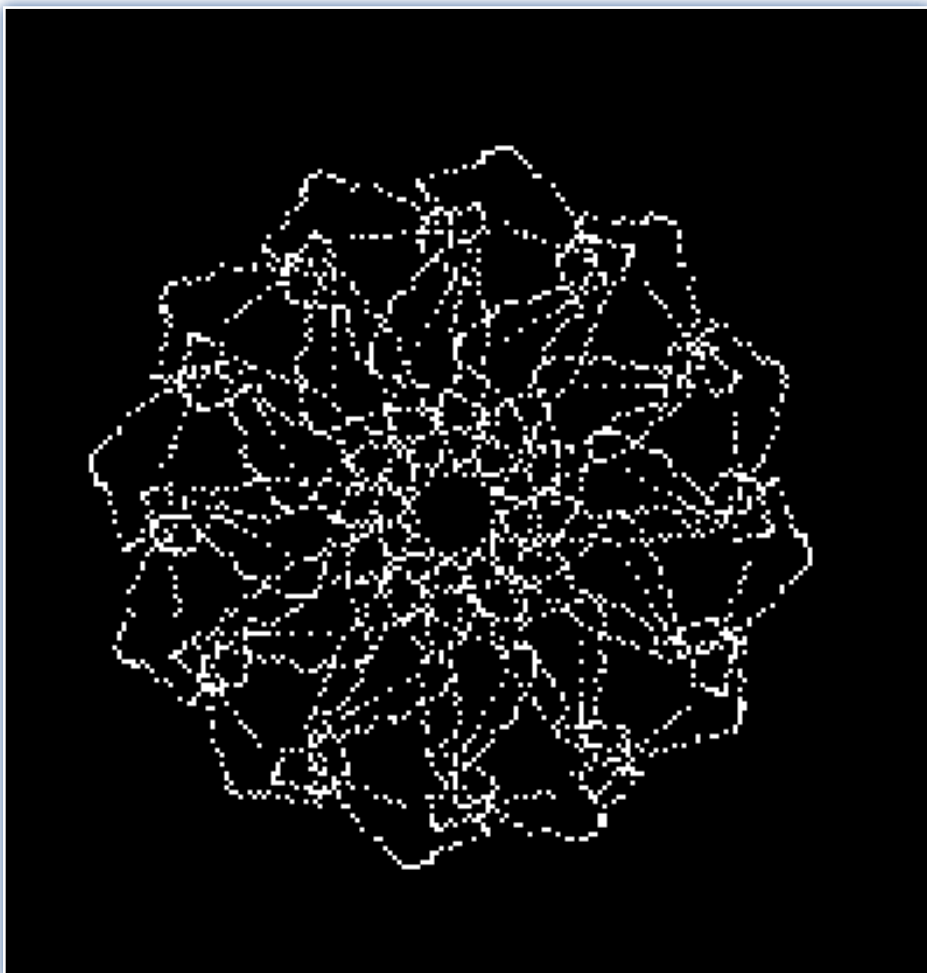
Można dodać automatyczne poruszanie punktem głównym, uzyskując naprawdę interesujące obrazki. Uzyskany efekt wyglądający jak poniżej.

Dodatkowo można eksperymentować z ilością punktów obracanych wokół środka. Jest wiele różnych możliwości.



Zachęcam do własnych eksperymentów.

Axel



Wektory

na komodory

część 2

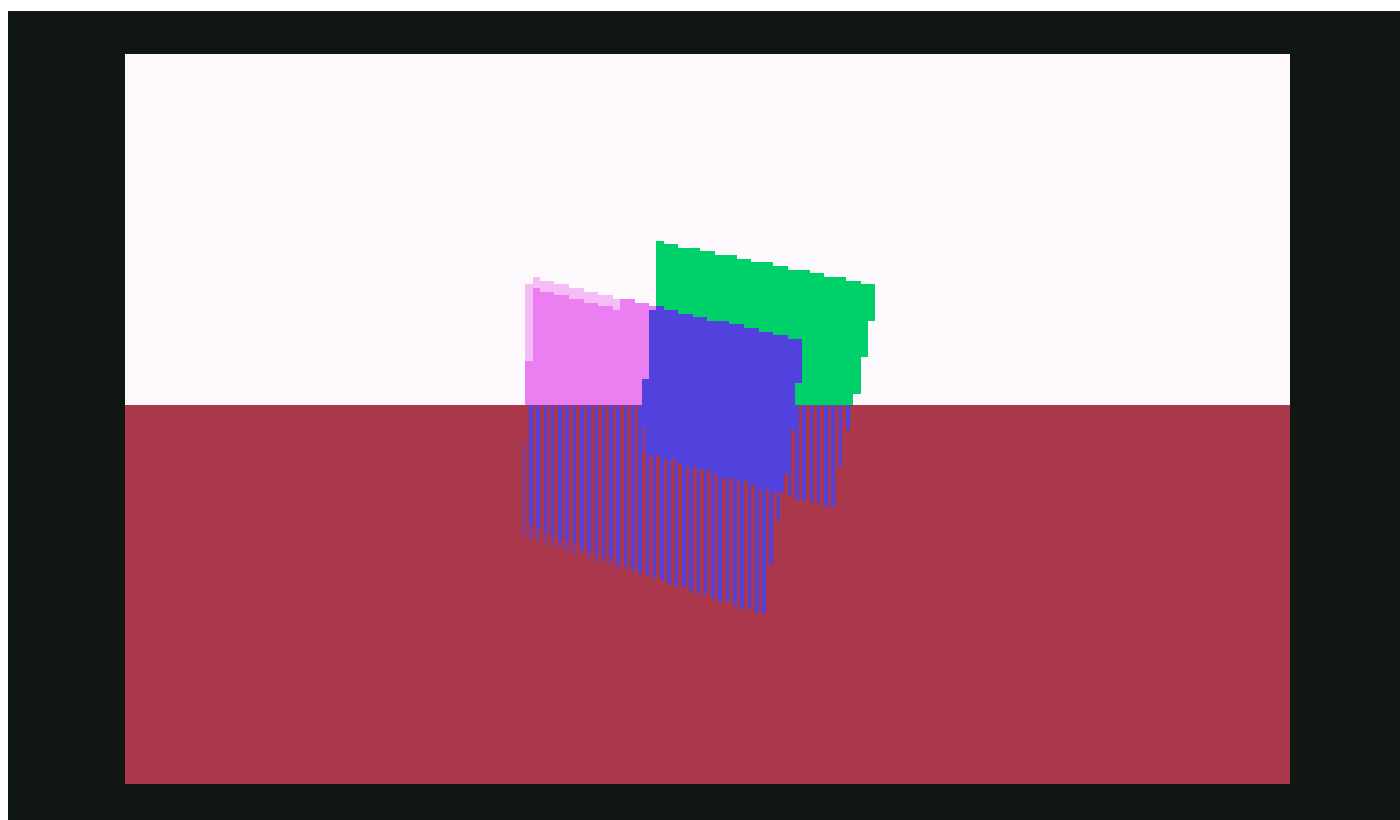
Dzięki dla wszystkich za uwagi na forum i na priv – największą aktywnością wykazał się o dziwo hardware-man Kisiel, dodał też ciekawe pomysły bazujące oczywiście na ulepszeniach sprzętowych. Stwierdził, że nie może tylko sklonować części mechanicznych, a ja na to, że z pewnością coś wymyśli ;-)

Dear RED: Proszę dajcie obrazki przynajmniej 2 razy większe niż poprzednio. (Mówisz i masz ;-) - przypisek redakcji).

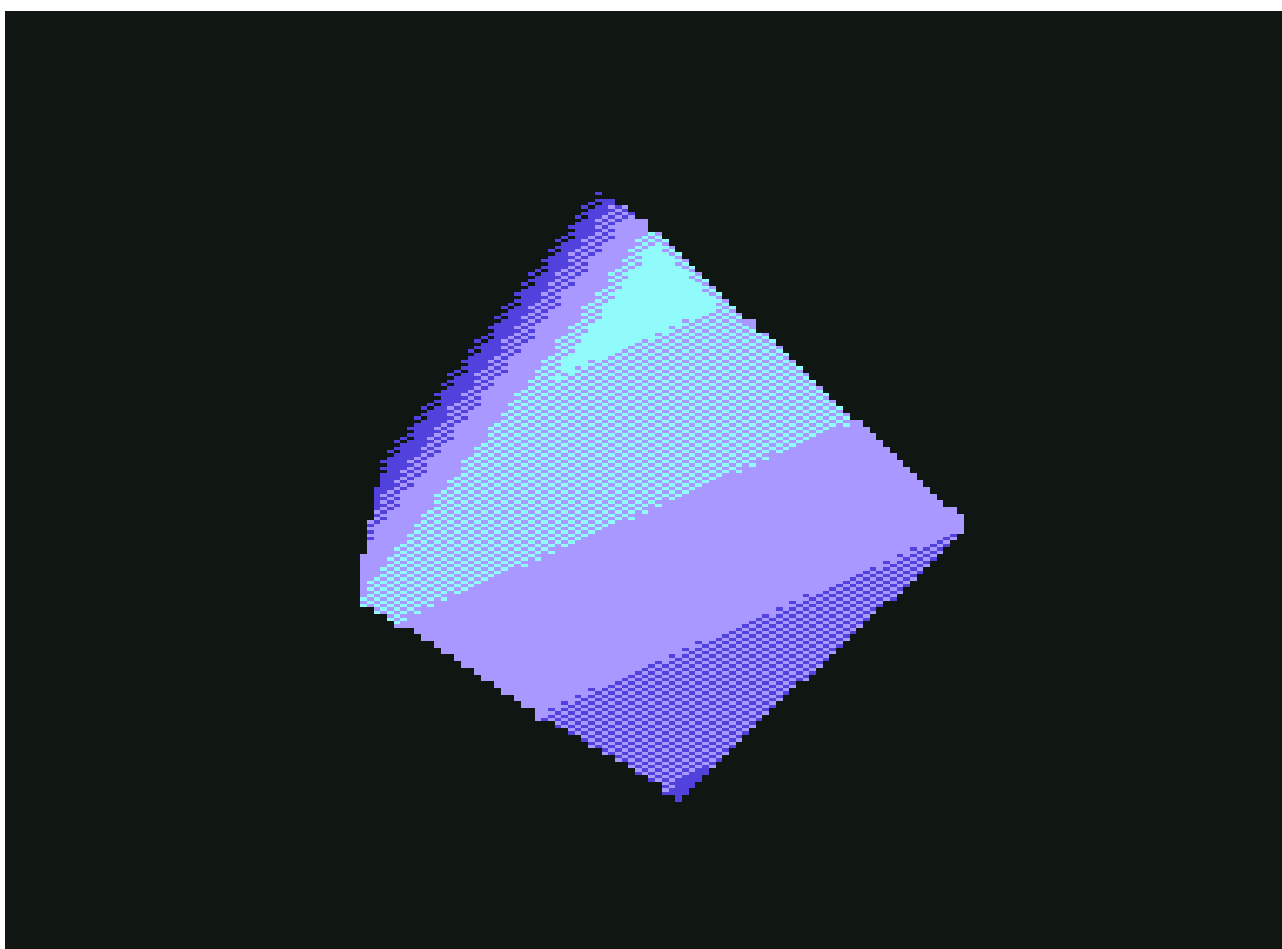
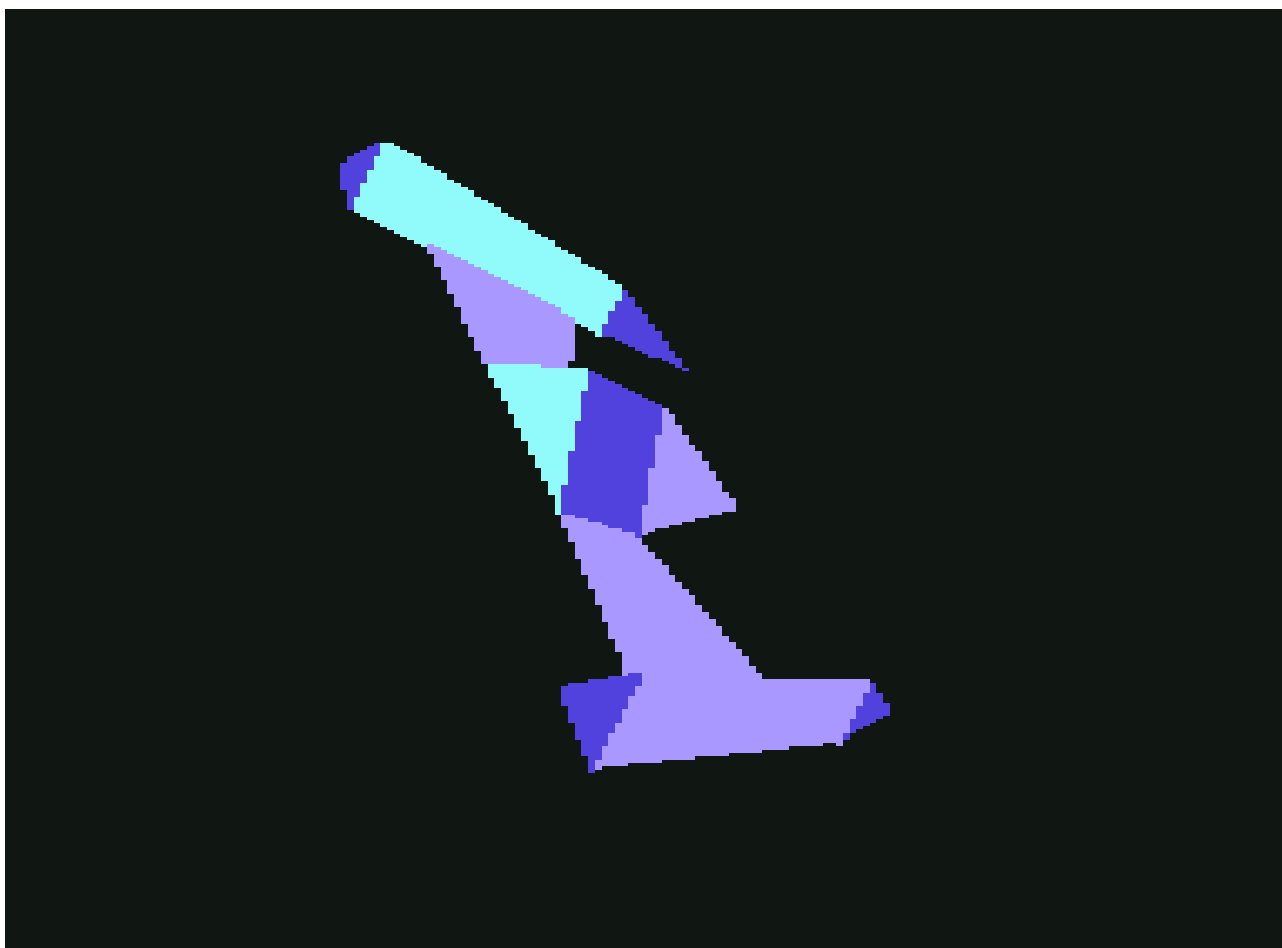
Cóż biorąc pod uwagę częstotliwość ukazywania się maga dzisiaj ostatnia część kursu ale będzie długa...

Dobra zaczynamy aby nie tracić czasu. Poniżej to co będzie uwieńczeniem kursu.

Glenz 2 ścian - są 2 kolory a tam gdzie ściany się nakładają jest trzeci - oczywiście calculated in drive ;).

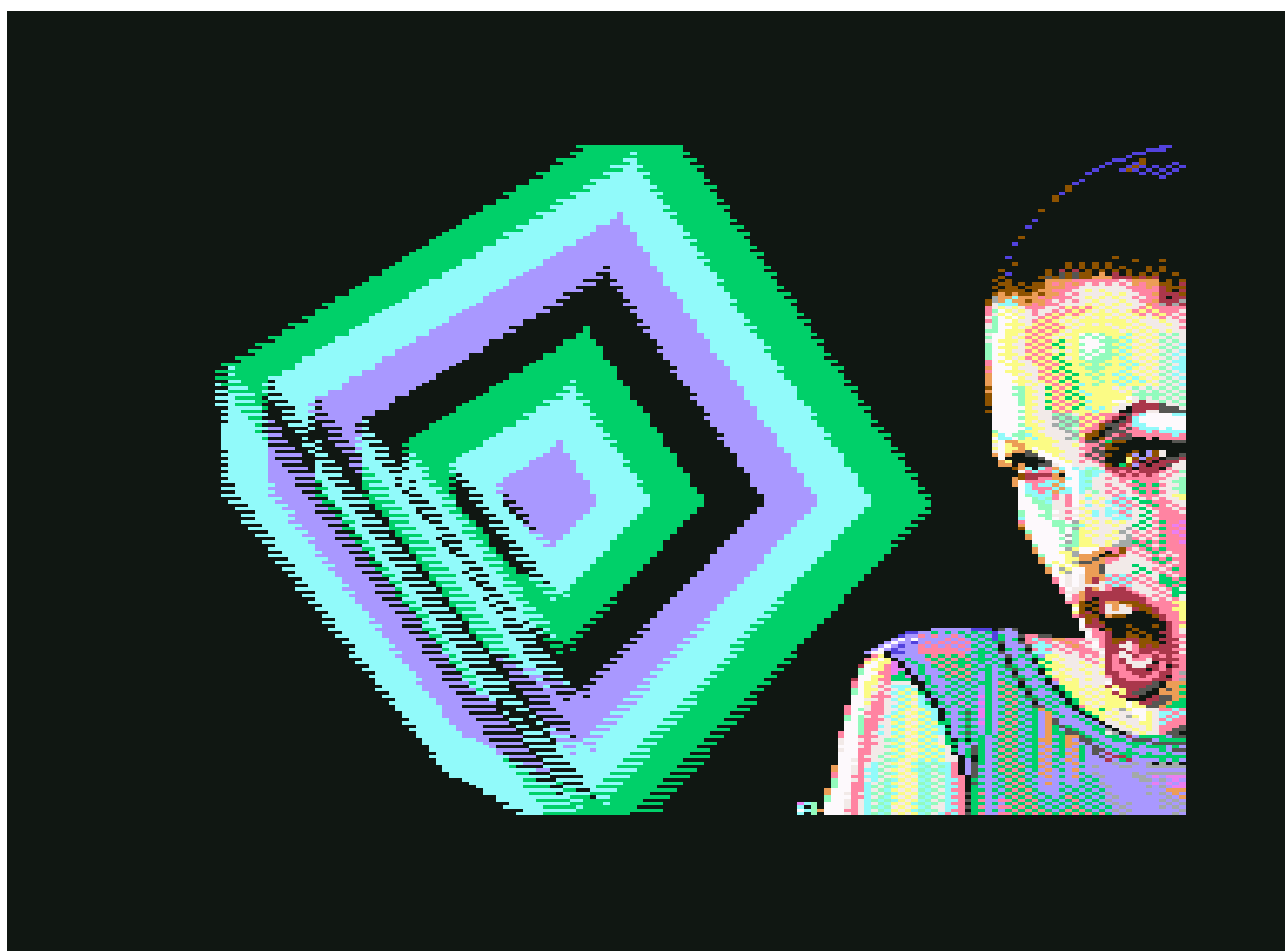
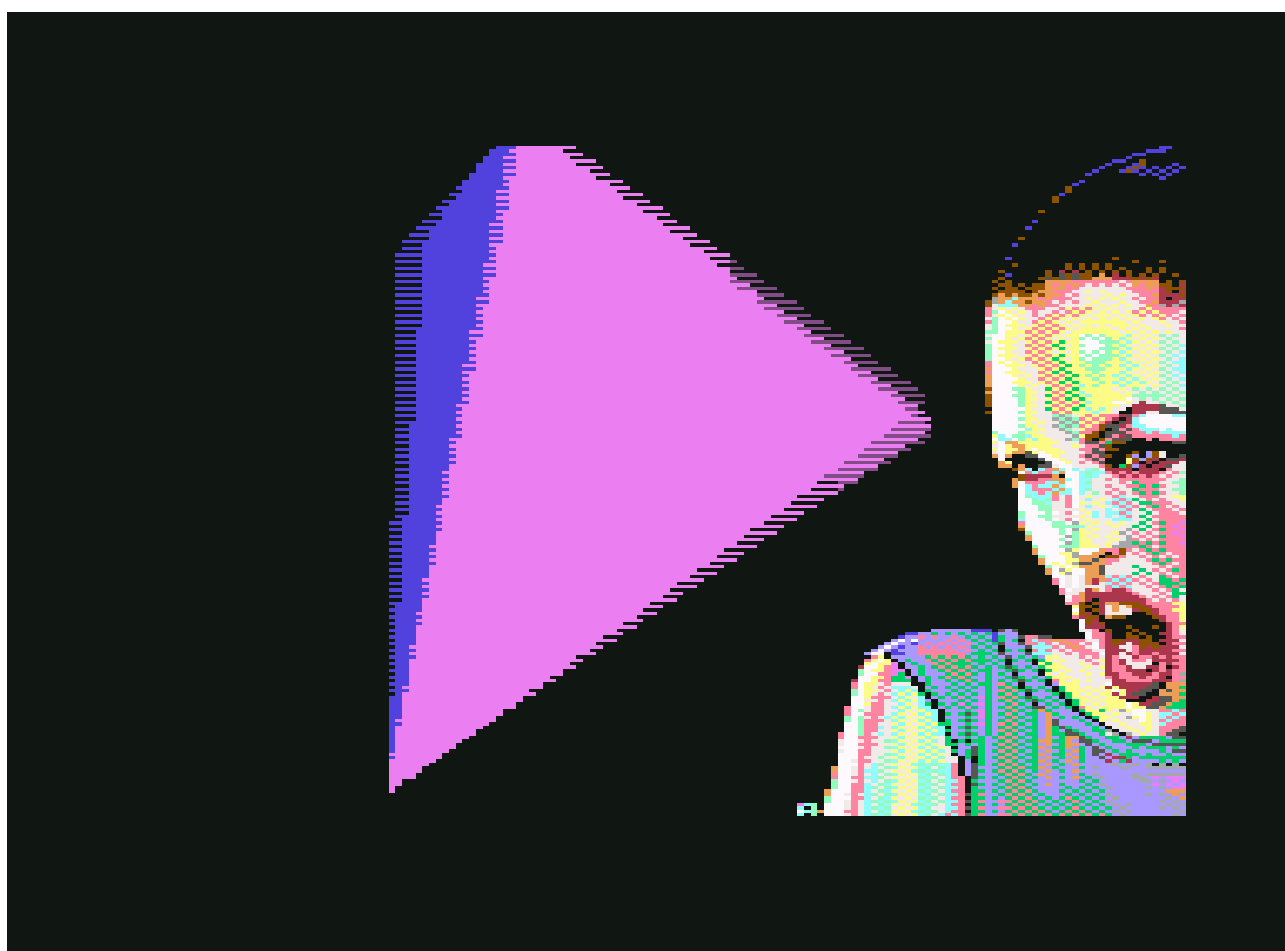


Zobaczmy ogromne wektory z Radia Napalm



A dojścia do TAAAKICH wektorów Wam życzę – REAL (2004)





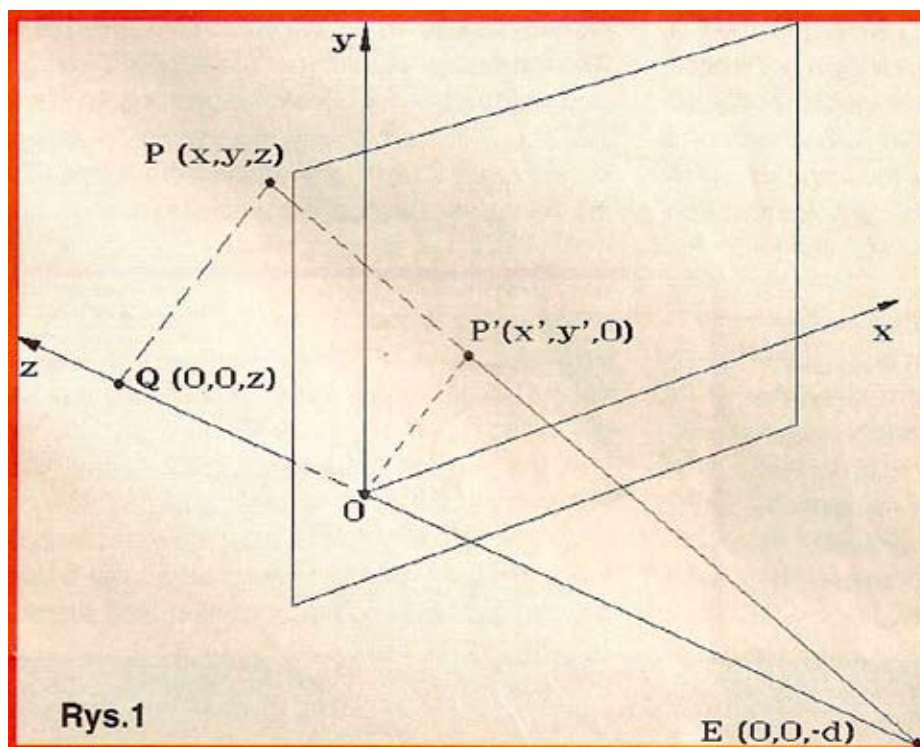
Oki – zakończyliśmy na mnożeniu ze znakiem liczb 8-bitowych i zapewniam Was, że dla naszej wektorówki jest ono wystarczające. Nie wiele więcej nam trzeba oprócz dalszych sztuczek wspomagających algorytm, ponieważ nie mamy ani rozkazów dzielenia czy mnożenia, ani też koprocatora jaki bywał (nie zawsze) już od 80x386 w PC i wykonywał operacje zmienno-przecinkowe.

Trochę rozważań matematycznych. Nie bójcie się, są to informacje ze szkoły podstawowej – przynajmniej ja miałem w szkole podstawowej funkcje sinus i cosinus oraz zasadę przemienności mnożenia.

Przy okazji - w rozmowie padła uwaga, że w kursie powinna być także informacja o grafice 2D – przy czym 2D jest jakby częścią 3D, ale na specjalne życzenie nie będzie to domyślnie, czy jakby zakamuflowane w treści – szybciej przejdziemy przez 2D.

Zaraz będą wzory na obroty punktu w trójwymiarowym układzie kartezjańskim. Od dwuwymiarowego różni się tym, że ma jedną oś więcej – Oś „Z”. Teoretycznie nawet jej nie widzimy bo jest ona skierowana prostopadle do obserwatora i widzimy punkt, a punkt jak wiemy jest nieskończenie mały więc go nie widzimy – pozostaje wyobraźnia ;-). Ta ZET właśnie to nasz trzeci wymiar – głębia.

Mniej więcej tak to wygląda:



To obrazek z C&A 95/08 strona 13, artykuł Przemka Cieślaka o tytule „Programowanie grafiki wektorowej” (nomen-omen). Widać tu trzy osie i rzutowanie perspektywiczne punktu na płaszczyznę. Póki co SKUPMY SIĘ NA TRZECH OSIACH.

Oj... teraz czeka mnie cała masa tłumaczenia.

Pamiętamy układ kartezjański 2D – tam każdy punkt miał współrzędne (X;Y), logiczne jest, że jak doszła nam trzecia oś, to teraz każdy punkt ma 3 współrzędne (X;Y;Z).

Poniżej wzory z codebase na obroty punktu wokół trzech osi tego układu:

Rotation about the x axis:

$$\begin{aligned}x' &= x \\ y' &= (\cos(xangle) * y) - (\sin(xangle) * z) \\ z' &= (\sin(xangle) * y) + (\cos(xangle) * z)\end{aligned}$$

Rotation about the y axis:

$$\begin{aligned}x' &= (\cos(yangle) * x) + (\sin(yangle) * z) \\ y' &= y \\ z' &= -(\sin(yangle) * x) + (\cos(yangle) * z)\end{aligned}$$

Rotation about the z axis:

$$\begin{aligned}x' &= (\cos(zangle) * x) - (\sin(zangle) * y) \\ y' &= (\sin(zangle) * x) + (\cos(zangle) * y) \\ z' &= z\end{aligned}$$

Przerażające?

Ja tu widzę nadal TYLKO 12 MNOŻEŃ 3 dodawania i 3 odejmowania.

Dalej widzę we wzorach wartości sinusów i cosinusów – czyli nadal **NIC SKOMPLIKOWANEGO**.

POTEORETYZUJMY

2 DIMENSION

Jeżeli chcielibyśmy obracać punkt wokół jednej osi, wówczas wystarczy wykonać raptem cztery mnożenia, jedno dodawanie i jedno odejmowanie. Będzie to nasze 2D...

FUNKCJE SINUS I COSINUS

Te dwie funkcje są funkcjami okresowymi co oznacza, że co okres (...) ich wartości się powtarzają. Na dalsze rozważania weźmy pod uwagę fakt, że te funkcje są przesunięte w fazie między sobą o ... (ile?) stopni co oznacza, że można odczytywać ich wartości z tej samej tabeli, należy tylko działać z przesunięciem... (ilu?) stopni – chodzi tutaj szczególnie o oszczędność miejsca w RAM stacji dysków... Czyli tak naprawdę wystarczy nie tylko jedna tabelka ale jej **POŁOWA**, natomiast znakiem jest offset odwołania (radzę przypomnieć sobie funkcje sinus i cosinus dla pełnego zrozumienia). Wystarczy odwoływać się do tabelki modulo część (ile?) okresu i można mieć w tabelce liczby unsigned – przypominam, że znakiem jest pozycja (wielkość przesunięcia) w tabeli. To z kolei przy każdym mnożeniu zaoszczędzi nam czas na konwersji $U \Leftrightarrow S$ – przy ośmiu wierzchołkach sześcianu i obrotu w trzech osiach to jest 96 mnożeń !

Wartość sinus 30 stopni wynosi 0.5 (1/2)
Sinus 10 stopni to wartość w przybliżeniu 0.1736
Nasze mnożenie jest na liczbach całkowitych...
Hmmm... ZONK???

Niezupełnie – pomyślmy o takiej prostej zasadzie - **PODKREŚLAM ZE SZKOŁY PODSTAWOWEJ** – przemienności mnożenia:

$$3*7*5 = 7*5*3$$

$$\begin{aligned}\text{albo} \\ 3*7 &= 7*3\end{aligned}$$

dalej
 $3*7*1000/1000 = 3*7*1$ bo $1000/1000 = 1$ czyli docelowo $3*7$ – to kluczowy moment rozważań, bo jak pamiętamy o przemienności mnożenia to:

$$3*1000*7/1000 = \text{NADAL } 3*7 (!!!)$$

OTO NASZE REMEDIUM !!!

Podpowiem jeszcze, że jak pomnożymy nasze wartości przez jakąś potęgę liczby 2 np. 256 to niesłychanie łatwo będzie nam ją na końcu dzielić, ponieważ odwołamy się tylko do starszego bajtu wyniku (w przypadku 256)!

TYCH PROSTYCH OPERACJI UCZYLIŚMY SIĘ W SZKOLE PODSTAWOWEJ!!!

Xangle, Yangle, Zangle to żądany kąt obrotu wokół wybranej osi.

Jeżeli weźmiemy sobie z 360 stopni wartości sinusa w rozdzielczości 360/256 to będziemy mogli obracać nasz punkt co około 1.5 stopnia, co jest w zupełności wystarczające (a nawet aż nadto) do stworzenia płynnej animacji wektorowej.

A więc oto programik w BASIC'u generujący naszą tabelkę, z której wykorzystamy tylko połowę! – reszta to kontrola programowa.

```
10 for i = 0 to 255
20 d = int(sin((2*PI/256)*i)*256)
22 if d < 0 then d = not d
30 printd
40 poke 4096+i,d
99 next
```

Dobrze. Teraz zastanówmy się co się stanie, jak zaczniemy obracać wybrany punkt właśnie wokół osi Z... Powinien zatoczyć okrąg – zatem możemy sprawdzić, czy podane wzory są poprawne i czy wnioski z rozważań są właściwe – co trzeba zrobić?

Należy wybrany punkt obrócić wokół osi Z i wyświetlić jego pozycję na bitmapie



Poniżej listing programu realizujący naszą rotację:

```
*= $0801
;-----
.BYTE $0B,$08,$90,$06,$9E,$32
.BYTE $30,$34,$39,$00,$A0,$00
;-----
;SIMPLE 2D ROUTINE
;(C) BY WEGI
;SPECIAL FOR C&A FAN - SITE: CA-
FAN.PL
;-----
;-----
STOREPLOT = $D3
SCREEN = $2000
;---
SEI
CLD
LDX #$FB
TXS
LDA #$37
STA $01
JSR $FDA3
```

```
JSR $FF5B

JSR SETTBADR
JSR INITGRAPH
SEI
;---
DRAW
JSR ROTATE
LDA #156
SEC
SBC XLAST
STA POZX
LDA YLAST
CLC
ADC #100
STA POZY

JSR PLOT
LDA ANGZ
BNE DRAW
LDA MAINY
SEC
SBC #15
BCS STOREY
LDA #100
STOREY
STA MAINY
LDA #SEF
CMP $DC01
BEQ *-3
JMP DRAW
```

```
;-----
INITGRAPH

; LDA #$06
; STA 53280

LDA 53272
ORA #$08
STA 53272
```

```
LDA 53265
ORA #32
STA 53265
```

```
LDX #$00
LDA #246
FILL1
```

```
STA $0400,X
STA $0500,X
STA $0600,X
STA $06F8,X
INX
BNE FILL1

LDX #$20
STX $FC
LDY #$00
STY $FB
LDA #$00
```

```
FILL2 STA ($FB),Y
INY
BNE FILL2
INC $FC
DEX
BNE FILL2
RTS
SETTBADR
LDX #$00
LDA #>SCREEN
STX $FB
STA $FC
SETTB2
LDA $FB
STA TBADLO,X
LDA $FC
STA TBADHI,X
```

```
LDA $FB
CLC
ADC #$40
STA $FB
```

```
LDA $FC
ADC #$01
STA $FC
INX
CPX #25
BCC SETTB2
RTS
```

```
;-----
TBBIT
.BYTE %10000000
.BYTE %01000000
.BYTE %00100000
.BYTE %00010000
.BYTE %00001000
.BYTE %00000100
.BYTE %00000010
.BYTE %00000001
```

```
;---
TBADLO
.BYTE 0,0,0,0,0
.BYTE 0,0,0,0,0
.BYTE 0,0,0,0,0
.BYTE 0,0,0,0,0
.BYTE 0,0,0,0,0
```

```
;---
TBADHI
.BYTE 0,0,0,0,0
.BYTE 0,0,0,0,0
.BYTE 0,0,0,0,0
.BYTE 0,0,0,0,0
.BYTE 0,0,0,0,0
```

```
;-----
;-----
POZX .BYTE 0,0
POZY .BYTE 0,0
```


PLOT

```
LDA POZY
LSR A
LSR A
LSR A
TAX
```

```
LDA POZY
AND #$07
TAY
```

```
LDA POZX
AND #$F8
CLC
ADC TBADLO,X
STA STOREPLOT
```

```
LDA TBADHI,X
ADC POZX+1
STA STOREPLOT+1
```

```
LDA POZX
AND #$07
TAX
```

```
LDA (STOREPLOT),Y
EOR TBBIT,X
STA (STOREPLOT),Y
```

RTS

```
;-----
;-----
MNOZNA .BYTE 0
MNOZNIK .BYTE 0
ANGLEZ .BYTE 1
DANA1 .BYTE 0
;-----
MAINX .BYTE 0
MAINY .BYTE 100

XPOINT .BYTE 0
YPOINT .BYTE 0

ANGZ .BYTE 0
;-----
SINZ .BYTE 0
SINZSIGN .BYTE 0
COSZ .BYTE 0

COSXSIGN .BYTE 0
COSYSIGN .BYTE 0
COSZSIGN .BYTE 0
;-
XLAST .BYTE 0
YLAST .BYTE 0
;-
SIGN .BYTE 0
;-----
;-----
```

PROCKA

```
BPL LP1
EOR #$FF
CLC
```

```
ADC #$01
LP1 STA MNOZNA
STY MNOZNIK
```

```
LDA #$00
LDX #$08
```

MNOZE

```
ROR MNOZNIK
BCC LP2
CLC
ADC MNOZNA
```

```
LP2 ROR A
DEX
BNE MNOZE
```

```
; ROR MNOZNIK
;NOT IMPORTANT - HI BYTE IS
RESULT
```

```
BIT SIGN
BPL LP3
```

```
EOR #$FF
CLC
ADC #$01
```

LP3

RTS

```
;-----
;-----
ROTATE
;-----
; COSINUS Z & SIGN COSINUS Z
;-----
FINDSINE
;---
LDA #$00 ;SIGN +
STA COSZSIGN

LDA ANGLEZ
CLC
ADC ANGZ

STA ANGZ

CMP #$40 ;COS ANGLE Z
BCC FINDS3 ;SIGN +
CMP #$C0
BCS FINDS2 ;SIGN +
DEC COSZSIGN ;SIGN -
FINDS2
CLC ;ADD QUARTER
PERIOD
FINDS3
ADC #$40 ;TO COS.
AND #$7F ;MODULO HALF
PERIOD
TAY
LDA SINE,Y
```

STA COSZ

```
;-----
;--- SINUS Z
;-----
LDA ANGZ ;ANG IS ALSO SIGN
AND #$7F ;LIKE BEFORE
TAY
LDA SINE,Y
STA SINZ
;-----
LDA MAINX
STA XPOINT
LDA MAINY
STA YPOINT
;-----
;--- X LAST
;-----
LDY SINZ
LDA YPOINT
EOR ANGZ
STA SIGN
LDA YPOINT ;SIN(ANGZ)*Y
JSR PROCKA
STA DANA1

LDY COSZ
LDA XPOINT ;COS(ANGZ)*X
EOR COSZSIGN
STA SIGN
LDA XPOINT

JSR PROCKA
SEC
;COS(Z)*X - SIN(Z)*Y

SBC DANA1 ;SUBSTR.
STA XLAST
;-----
;--- Y LAST
;-----
LDY SINZ
LDA XPOINT
EOR ANGZ
STA SIGN
LDA XPOINT ;SIN(ANGZ)*X
JSR PROCKA
STA DANA1

LDY COSZ
LDA YPOINT ;COS(ANGZ)*Y
EOR COSZSIGN
STA SIGN
LDA YPOINT

JSR PROCKA
CLC ;SIN(Z)*X + COS(Z)*Y
ADC DANA1 ;SUM

STA YLAST ;YLAST
;-----
RTS
;-----
```

```

;-----
SINE
.BYTE $00,$06,$0C,$12,$19,$1F
.BYTE $25,$2B,$31,$38,$3E,$44
.BYTE $4A,$50,$56,$5C,$61,$67
.BYTE $6D,$73,$78,$7E,$83,$88
.BYTE $8E,$93,$98,$9D,$A2,$A7
.BYTE $AB,$B0,$B5,$B9,$BD,$C1
.BYTE $C5,$C9,$CD,$D1,$D4,$D8
.BYTE $DB,$DE,$E1,$E4,$E7,$EA
.BYTE $EC,$EE,$F1,$F3,$F4,$F6
.BYTE $F8,$F9,$FB,$FC,$FD,$FE
.BYTE $FE,$FF,$FF,$FF,$FF,$FF
.BYTE $FF,$FF,$FE,$FE,$FD,$FC
.BYTE $FB,$F9,$F8,$F6,$F4,$F3
.BYTE $F1,$EE,$EC,$EA,$E7,$E4
.BYTE $E1,$DE,$DB,$D8,$D4,$D1
.BYTE $CD,$C9,$C5,$C1,$BD,$B9
.BYTE $B5,$B0,$AB,$A7,$A2,$9D
.BYTE $98,$93,$8E,$88,$83,$7E
.BYTE $78,$73,$6D,$67,$61,$5C
.BYTE $56,$50,$4A,$44,$3E,$38
.BYTE $31,$2B,$25,$1F,$19,$12
.BYTE $0C,$06,$00
;-----

```

Poprzedni program z mnożeniem, zubożony został o konwersję hex i bcd, a dodana została obsługa trybu graficznego – stawianie plota – oraz „czary mary” z sinusami – to sobie przeanalizujcie... Szczerze mówiąc – niewiele zmian, a widać już efekty.

Po obliczeniach ważna jest translacja na współrzędne ekranowe, ponieważ wynik otrzymujemy ze znakiem -, a na ekranie ujemne współrzędne nie istnieją. Do własnych rozważań polecam również fakt, że w układzie współrzędnych wielkość Y wzrasta w górę, a na ekranie w dół – jak będzie wyglądał zatem nasz wektorowy obiekt?? Tyle na temat translacji z układu współrzędnych do współrzędnych ekranowych. Nie ma się nad czym rozwodzić... Jak widać wzory nie kłamią i program daje radę :D

Prawdę mówiąc to już niewiele brakuje tutaj do stworzenia wektorówki i co bardziej zaawansowani programerzy pewnie by sobie poradzi. Wystarczy tylko doroobić pętlę dla większej ilości wierzchołków, obracać w trzech osiach oraz RZUTOWAĆ PERSPEKTYWICZNIE naszą bryłę. Potem musimy poznać algorytm rysowania linii i wypełniania. Podczas obliczeń warto sprawdzać, czy obracamy we wszystkich osiach, jeżeli nie, to mamy zaoszczędzone 30% obliczeń na każdą oś – pamiętajmy o tym!

PERSPEKTYWA

Istnieją wzory na perspektywę i można sobie o nich rozprawiać, a ja powiem tak: jeżeli staniami na środku prostej asfaltowej drogi i popatrzymy na wprost przed siebie to nasza droga, pomimo że przez całą długość ma tą samą szerokość, to czym dalej patrzymy wydaje się nam ona coraz węższa – to jest właśnie per-

spektywa – obiekty im bardziej oddalone są od obserwatora, tym mniejsze się wydają. Jak spojrzemy na pierwszy obrazek to widzimy dwie ściany, które są konstrukcyjnie tych samych wymiarów, natomiast ściana z tyłu (zielona) jest widocznie mniejsza – czyli wszystko gra. Oglądamy naszą bryłę z perspektywy!

Wspomniałem na początku, że 90% obliczeń wektorów to właśnie mnożenie, co nie mijają się z prawdą, dalej są dodawania i odejmowania oraz dwa dzielenia dla uzyskania perspektywy z rzutu 3D na 2D. W niektórych wektorówkach widać źle dobraną perspektywę (odległość obserwatora) i widać, że te tylnie ściany są zbyt małe (nienaturalnie) i bardzo ostro schodzą się do środka krawędzie łączące przednią ścianę z tylną. Czasami na odwrót – tylna ściana jest niemal taka sama jak przednia i też nienaturalnie to wygląda.

Dobrze. Wzory na perspektywę:

$$X2D = X * d / (Z + d)$$

$$Y2D = Y * d / (Z + d)$$

Założenie: $d < (-Z)$ (albo $Z + d < 0$) aby uniknąć dzielenia przez 0

X, Y i Z to współrzędne wierzchołka obliczone po obrotach, a „d” to jest odległość obserwatora od niego. Zauważmy, że przy obliczeniach należy uwzględnić to, że jak Z jest ujemne to faktycznie Z rośnie, bo punkt oddala się od obserwatora. Te dwa proste wzory naprawdę urealniają naszą grafikę wektorową.

W pierwotnym zamyśle chciałem pokazać perspektywę z programowym dzieleniem jednakże pokażę ją z użyciem tablic, ponieważ będzie to zaprezentowanie odmiennych możliwości i cennego, szybkiego rozwiązania.

Perspektywa potrafi być zmorem dla kodera, jej dobranie, zooming (prescale) to naprawdę trzeba się w to wgrzyć, bo nie ma tutaj lekko. Dochodzi dzielenie ze znakiem liczby 16-bitowej przez niekoniecznie 8-bitową, może trochę większą. Istnieją różne sposoby, od tabel, poprzez dzielenie programowe, dzielenie z użyciem tablic pierwiastków (polecam codebase i C=hacking!!!). Użyte rozwiązanie jest o tyle ciekawe, że jest szybkie, wymaga 256 bajtowej tablicy „konwersji” naszego Z, natomiast jeszcze ciekawsze jest uzyskanie później zoomingu. Tablicę generujemy takim lub podobnym programem w BASIC:

```

10 bz=4096
20 d=45:z0=3:z=-128:dz=1
30 for i=0 to 255
40 q%=64*d/(64*z0-z):if
q%>127 then q%=127
50 poke bz+i,q%:z=z+dz
60 next

```

To jest akurat programik z C=Hacking #8, można samemu pokombinować swoją perspektywę, nikogo nie zmuszam, aby z niego korzystać. Można poeksperymentować z parametrami, ja je pozmieniałem oraz „przeoorowałem” tablicę, żeby skrócić kod przetwarzający owy Z wynik. Banalnie odczytuje się daną:

```

LDA Z
CLC
ADC #128
TAX
LDA TABELA,X
TAY

```

Wyjaśnienia wymaga dodanie 128 do naszego Z – otóż wynika to z faktu, że tak naprawdę im mniejsze jest Z, tym punkt jest dalej od obserwatora i w tym miejscu właśnie to uwzględniamy, ponieważ tabelka nie jest całkowicie „dostosowana” do naszego Z. W tym momencie w rejestrze Y mamy nasze „sprefabrykowane” Z, jest ono teraz liczbą bez znaku i może mieć wartość większą od 128, nie konwertujemy już jej do signed, natomiast drugi parametr do mnożenia (będzie to X albo Y po rotacjach) jest ze znakiem i to uwzględniamy w taki sposób:

```

LDA X3D
STA SIGN ;ZNAK DO MNOŻENIA
JSR SIGNEDMULTIPLY

```

Normalnie wcześniej przed TAY sprawdzałoby się znak i konwertowało nasze Z do U w razie potrzeby, potem eorowałoby się znak (SIGN), żeby wiedzieć jaki będzie wynik mnożenia, w tym przypadku znak będzie z naszego X3D tylko, jak mówiłem – Z może być większe od 128 ale jest bez znaku.

Dobrze po SIGNEDMULTIPLY mamy w akumulatorze wartość ze znakiem naszego X2D – czyli po perspektywie i teraz **najciekawsza rzecz**, jak łatwo można zrobić prescaling tej liczby do zoom’a. Wiadomo, że można zrobić najazd i odjazd obiektu, a nie tylko w jednakowej wielkości obracać obiekt. Wyobraźmy sobie, że tylko dzięki zwykłemu mnożeniu można uzyskać AŻ 255 stopni podziału danej liczby 8-bitowej ze znakiem lub bez – w jaki sposób?

Aby to zrozumieć, należy zobaczyć jak wygląda wynik mnożenia dwóch liczb 8-bitowych. Wynik jest w starszym i młodszym bajcie – zajmijmy się TYLKO STARSZYM BAJTEM WYNIKU – przykładowe wartości starszych bajtów wyniku z mnożenia dwóch 8-bitowych liczb:

```

$FF * $FF = $FE
$FF * $F0 = $EF
$FF * $81 = $80
$FF * $41 = $40
$FF * $11 = $10

```


dalej

```
$90 * $FF = $8F
$90 * $F0 = $87 !!! -> * 15/16tych
$90 * $81 = $48
$90 * $40 = $24
$90 * $10 = $09 -> * 1/16ta
```

Widać tu, że liniowo zmniejszamy naszą wielkość i mamy aż 255 możliwych stopni podziału (co jest aż nadto). Nie problem podzielić sobie liczbę przez 2 czy 8 = czyli razy 0.5 czy 0.125 natomiast podzielić ją przez 3, 5 czy 7, 11 nie jest już takie łatwe, a tu przy okazji ZWYKŁEGO mnożenia możemy preskalować daną liczbę!! Dokładnie w taki sposób zrobimy PŁYNNEGO ZOOM'A okresowo zwiększając bądź zmniejszając nasz „Z-FACTOR”. Proporcje są jak najbardziej zachowane i wszystko jest poprawne.

Myślę, że format liczb szesnastkowych nasręcza wiele niedogodności i rzadko mamy możliwość cieszyć się z jego właściwości i... to jest akurat ten moment ;-)

Wróćmy do naszej perspektywy:

```
LDA TABELA,X
TAY ; Spreparowane Z – MNOŻNIK U
      (unsigned)
LDA X3D ; MNOŻNA S(igned)
STA SIGN ;ZNAK DO MNOŻENIA
JSR SIGNEDMULTIPLY
```

Mamy w akumulatorze X2D – po perspektywie, wystarczy teraz jak poprzednio załadować Y wartością preskalingu i wykonać kolejne mnożenie. Zasady jak poprzednio – zoom (Y) jest U, a akumulator (X2D) jest ze znakiem, więc:

```
LDY ZOOM
STA SIGN ;ZNAK DO MNOŻENIA
TAX ;ODTWORZENIE ZNACZNIKA „N”
W REJESTRZE STANU PO LDY ZOOM
JSR SIGNEDMULTIPLY
```

Teraz mamy wartość ze znakiem naszego X po perspektywie i preskalingu. TAX nie jest wymagane do mnożenia, tylko ustawia potrzebny do niego znacznik „N”. Gwoli ścisłości dodam, to co pisałem poprzednio, że teraz nasze X trzeba TYLKO „przerobić” na współrzędne ekranowe – pamiętamy, że na bitmapie nie ma ujemnych wartości. Ponieważ operujemy na matrycy 128x128 pixeli musimy dodać do X połowę tej wielkości, co nie jest już żadną trudnością:

```
CLC
ADC #64
```

Jeżeli operujemy na znakowej bitmapie w

trybie multicolor (a operujemy), to musimy jeszcze „z uwagi na podwójną szerokość pikseli w X dla multicoloru, otrzymany wynik podzielić przez 2 co także jest banalne i zajmuje 2 cykle procesora

LSR

UWAGA!! – Dzielenie przez 2 dla trybu multicolor dotyczy TYLKO współrzędnych X. Jeżeli będzie to bitmapa czy chargen HIRES – LSR POMIJAMY.

JAK ZROBIĆ WEKTORY

Chciałbym zaplanować co trzeba zrobić aby wyświetlić wektory na C64:

1. Umieścić bryłę w układzie współrzędnych 3D i zapisać jej wierzchołki
 2. Obliczyć rotację w osiach wierzchołków bryły
 3. Obliczyć perspektywę dla danego wierzchołka (rzutowanie 3D na 2D)
 4. Dokonać translacji na współrzędne ekranowe
 5. Wierzchołki narysować łącząc je liniami aby utworzyć ściany
 6. Wypełnić ściany jeżeli nie robimy „drucików”.
- I w zasadzie taka to jest pętla do punktu 2.

Jeszcze jedna uwaga odnośnie obliczeń – jak widać poprzednio – cały czas obliczenia zaczynamy od naszej bryły tzn. nie ma czegoś takiego, że obróciliśmy punkt o 5 stopni i potem ten obrócony punkt obracamy o kolejne 5 stopni – **NIE – TAK NIE MOŻNA!** Do kolejnego obrotu nasz punkt obracamy o 10 stopni (15,20,25 itd.) i tak aż modulo 360 stopni – dla czego? Prosta odpowiedź : „**AKUMULACJA BŁĘDU**” – nasze tablice sinusów są pozaokrąglane, raz za razem narastałby nasz błąd obliczeń aż zaczęlibyśmy rysować wszystko poza naszym wektorem.

Do punktu 4 teorię przerobiliśmy wyłącznie z kodem (perspektywa, ze sporą dawką teorii), pozostał właściwie tylko punkt 5, ponieważ wypełnianie jest tak banalne, że nie ma się czym przejmować. Cały „wic” jest w punkcie piątym.

Ad. 5 i 6) W C64 ogólną praktyką jest tworzenie wektorów, gdzie jako bitmapa używany jest generator znaków w trybie multicolor. Dzieje się tak z dwóch powodów – po pierwsze działamy w polu matrycy 128x128 pixeli i jest ono mniejsze niż obszar bitmapy co daje mniejszy zakres operowania na RAM, oraz co ważniejsze – z uwagi na budowę chargena szybciej stawia się plota niż na bitmapie. Czyli da się zrobić wektory na bitmapie, tam też można działać w polu 128x128, natomiast szybciej jest na znakach.

RYŚOWANIE LINII

Teraz trochę od tyłu – wypełnianie to prosta operacja EOR z góry na dół po chargenie co każdy 8pixelowy pionowy pas. Aby jednak do-

brze wypełnić ściany potrzebny jest dobry algorytm rysowania linii. Linia jest rysowana na dwa sposoby – poziome linie są ciągłe, a pionowe są tylko kropkowane w miejscach przesko-ku (podziału). Poczytajcie także o algorytmie Brashenham. Aby połączyć ściany odpowiednimi liniami zapisuje się dane o naszej bryle w specjalnie do tego celu utworzonej strukturze, która pozwala takie informacje zachować. Wiemy o tym, że rozdzielczość każdego monitora jest skończona, a rozdzielczość obrazu generowana przez C64 nie należy do najwyższych. W dawnych czasach rysowania linii nauczył mnie Grabba (DZIĘKI!! :-)). On też wybił mi z głowy wypełnianie wektorów w poprzek (jeszcze raz DZIĘKI). Dostajecie dość powszechnie stosowaną i zaawansowaną procedurę rysowania linii po chargenie. Jest tu i automodyfikacja i optymalizacja. Popatrzmy na to:

```
;-----
LINIA1
      TXA
      ADC DY
      BCC LINIA1A

LINIA1B INY
      SBC DX
      BCS LINIA1B
LINIA1A TAX

CELL  = *+1
      LDA $E0

ADLN1 = *+1
      EOR $2000,Y

ADLN1A = *+1
      STA $2000,Y
;-----
```

Taki ciąg jest naszą linią, dla zwiększenia prędkości jest on przepisany wielokrotnie do RAM'u i tworzy jedną procedurę. To jednak nie wszystko. Aby narysować linię, należy zainicjować ją taką procedurą.

```
LDA COLOR
AND #$03
STA BITS+3
ASL A
ASL A
STA BITS+2
ASL A
ASL A
STA BITS+1
ASL A
ASL A
STA BITS
```

```
LDY MY1
STY Y1
LDY MY2
```

```

STY Y2
LDA MX1
STA X1
LDA MX2
STA X2

; LDA X2
SEC
SBC X1
BEQ DRWEX
BCS STOREDX
LDX X1
LDY X2
STX X2
STY X1

LDX Y1
LDY Y2
STX Y2
STY Y1
EOR #$FF
ADC #$01
STOREDX
STA DX
SEC
LDA Y2
SBC Y1
LDX X1
LDY X2
BCS VERLINE
EOR #$FF
ADC #$01
STA DY

LDA TBADLN+$80,X
STA ADLIN1+1

LDA TBADLN+$80,Y
STA VECTR1+1
RYSUJE
LDA TBADLN,X
STA ADLIN1
LDA TBADLN,Y
STA VECTR1

LDA #$60
LDY #$00
STA (VECTR1),Y
LDX #$FF
LDY Y1
SEI
CLC
ADLIN1 = *+1
JSR $1000
LDY #$00
LDA #$8A
STA (VECTR1),Y
RTS
VERLINE
STA DY

LDA TBADLN+$40,X
STA ADLIN1+1

```

```

LDA TBADLN+$40,Y
STA VECTR1+1

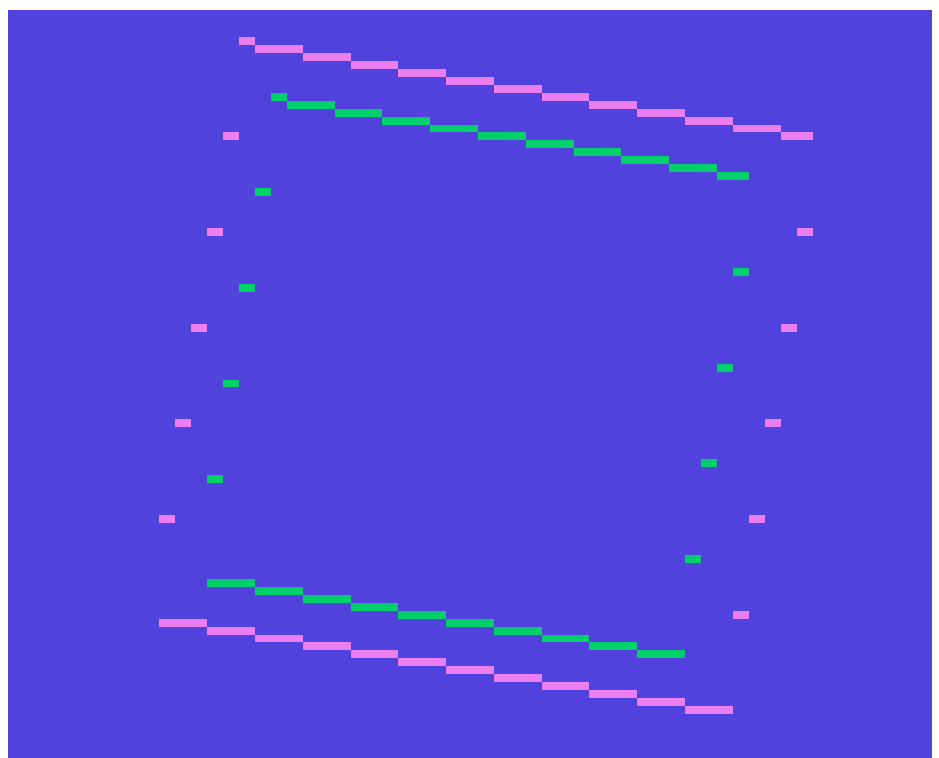
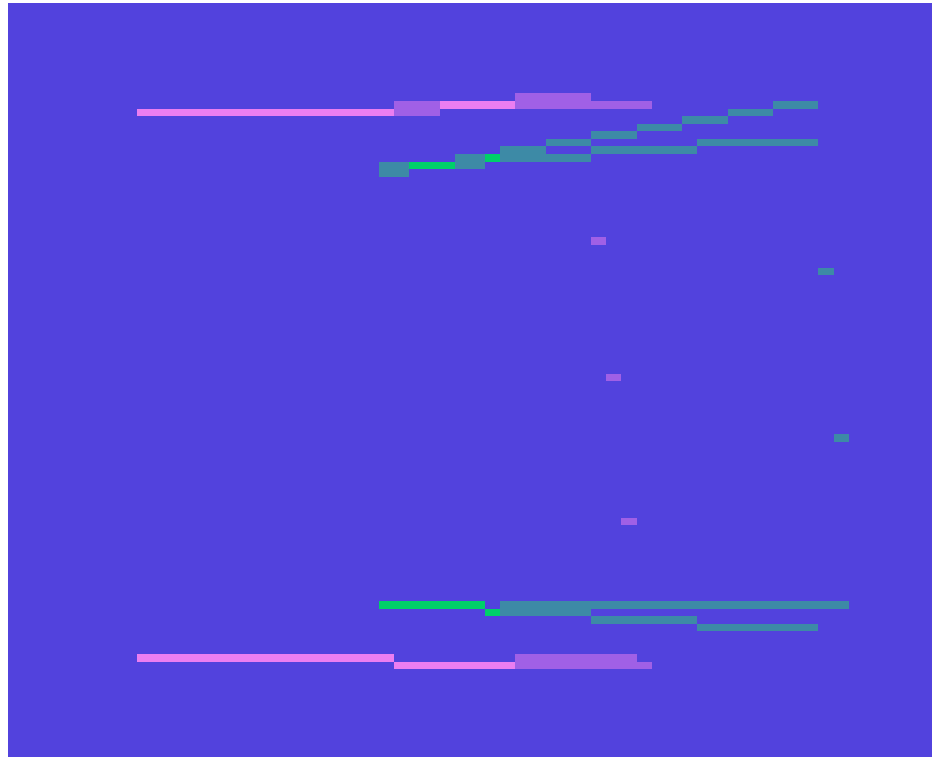
```

```
JMP RYSUJE
```

Na początku ustawiane są kolory plotów i przepisywane ich pozycje, później obliczana jest delta X i delta Y czyli przyrost, potem sprawdzane jest czy linia jest pionowa czy pozioma, następnie obliczany jest adres startowy i końcowy procedury w ciągu rysowania linii, potem wstawiany jest w potrzebne miejsce RTS i następuje skok do wyliczonego miejsca. Po skoku likwidowany jest nasz RTS. Cóż - czego

się nie robi dla szybkości ;-). Jak wspominałem linie pionowe i poziome są różnie rysowane – poziome są niejako rysowane ciągiem, a pionowe tylko w „miejscach przeskoku”. Jeżeli miałyby to być niewypełniane wektory (druciaki), to należy zmienić modyfikację w procedurze przygotowującej (preparującej) linię w ram.

Popatrzcie jak wyglądają linie w multicolorze dla filled vectors.



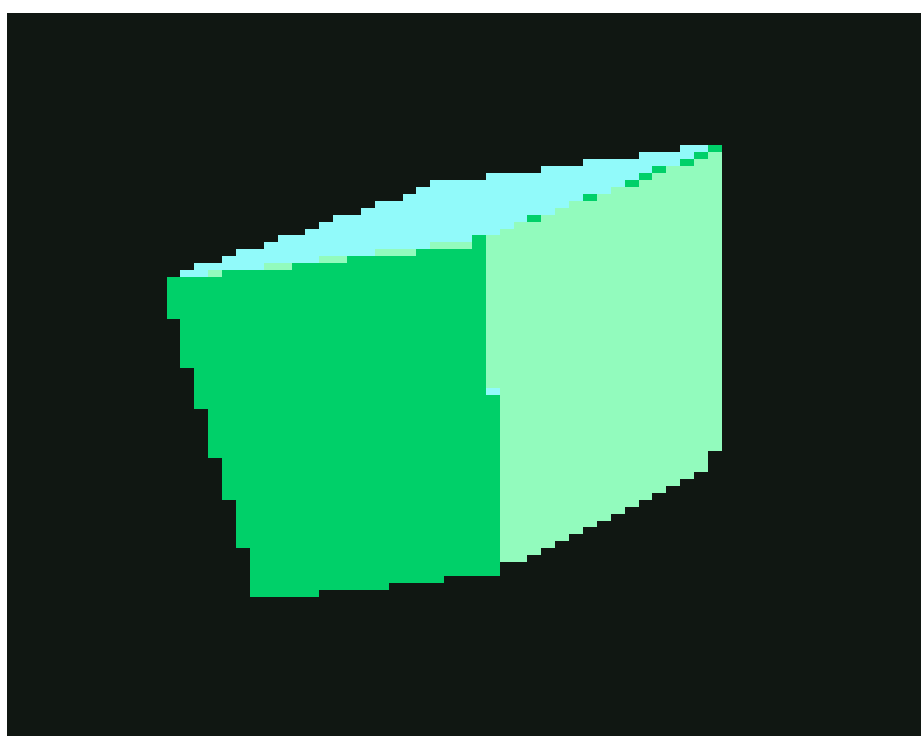
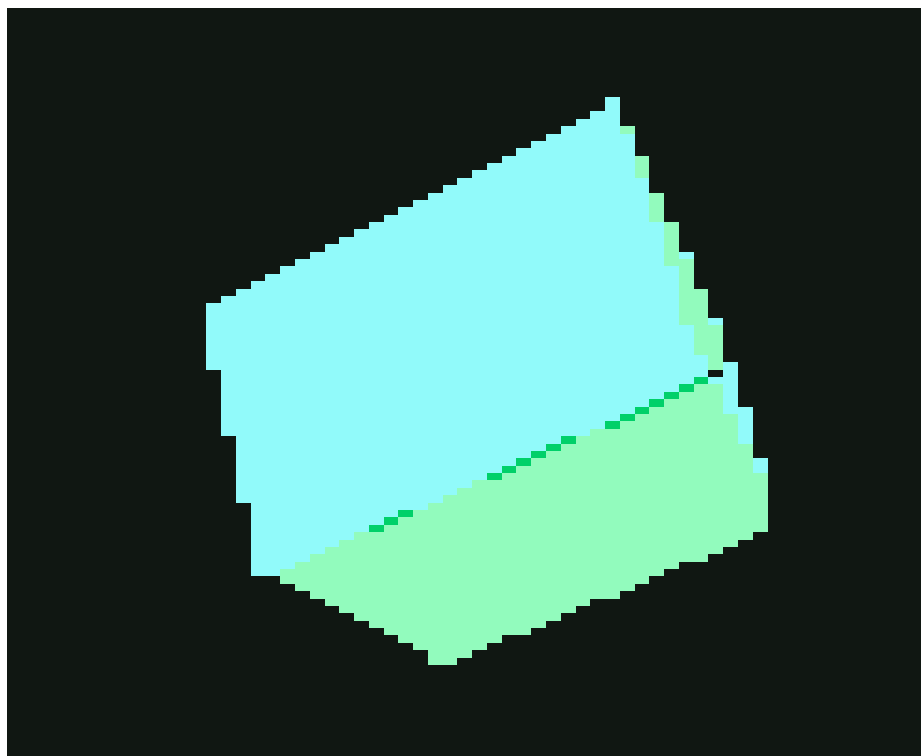
Co istotne – nie ma potrzeby rysowania PIONOWYCH linii – same się narysują przy wypełnianiu ;-)

Niesłychanie istotny jest kierunek rysowania przylegających do siebie krawędzi ścian, ponieważ jak się to zmieni, wówczas nasze ściany nie „poskładają się” właściwie i efekt będzie podobny do poniższego.

DALSZE TRICKI

Niebagatelna sztuczką jest zaprzęgnięcie do pracy drugiego procesora w stacji dysków,

ponieważ obliczenia trwają znacznie dłużej niż transmisja ich wyników. Sposób zainicjowany przez Polaków – byliśmy potęgą w kodowaniu na C64!!! Trick polega na umieszczeniu w stacji programu kalkulatora i na żądanie C64 wysłania gotowych obliczeń. W czasie gdy program w C64 zajęty jest rysowaniem linii, wypełnianiem, czyszczeniem bufora – stacja sobie spokojnie oblicza rotację. Jest to dobra filozofia, dla obrotu tylko 8-miu wierzchołków w trzech osiach i perspektywy trzeba, co KAŻDE przesunięcie, wykonać PONAD 100 MNOŻEŃ !!! Macie to ode mnie w gratisie :-)



PRACA WŁASNA

Nie podałem wektorów z eliminacją niewi-
docznych ścian - do tego musicie dojść sami. ...

Podpowiedź: Iloczyn skalarny – rozdział 10 książki A. Doligalskiego „Kurs assemblera dla początkujących”.

Wektory lekko szarpią – jako zadanie domowe doróbcie sobie podwójne buforowanie :-)

Jeszcze jedno – mając symetryczną bryłę, można bawić się w **mirrorong** i liczyć obroty np. w wypadku sześcianu dla pierwszych czterech wierzchołków, a potem znaleźć ich symetrię przez punkt 0;0;0 dla pozostałych czterech, później tylko perspektywa. Zaoszczędziłoby to połowę obliczeń...

PROGRAM OBROTÓW

Tu nasza stara procedura mnożenia zakłada, że Y jest bez znaku i Akumulator ze znakiem i w razie potrzeby konwertuje ACC. W Acc starszy bajt wyniku, co jak pamiętamy przy fixed poście pomnożonym przez 256 dzieli nam ponownie wynik przez 256. W zależności od znaku (zmienna sign) mnożenia konwertuje liczbę na S.

```
;-
PROCKA
    BPL LP1
    EOR #$FF
    CLC
    ADC #$01

LP1    STA MNOZNA
    STY MNOZNIK

    LDA #$00
    LDX #$08

LP1A   ROR MNOZNIK
    BCC LP2
    CLC
    ADC MNOZNA
LP2    ROR A
    DEX
    BNE LP1A

    BIT SIGN
    BPL LP3

    EOR #$FF
    CLC
    ADC #$01
LP3    RTS
```

Tutaj ustawienie kąta obrotów, ustalenie znaków dla sinusa i cosinusa (do wyniku mnożenia), pobranie wartości sinusa i cosinusa dla danego kąta. Przypominam, że tabelka sinusa to liczby bez znaku – fixed point pomnożone przez 256, znakiem dla sinusa i cosinusa jest

offset w tabeli. („...w pierwszej wszystkie są dodatnie, w drugiej tylko sinus...” :)))

```
;-----
ROTATES
    LDX #$02
;-----
; COSINUS ZYX & SIGN COSINUS ZYX
;-----
FINDSINE
;---
    LDA #$00    ;SIGN +
    STA COSXSIGN,X

    LDA ANGLEX,X
    CLC
    ADC ANGX,X

    STA ANGX,X

    CMP #$40    ;COS ANGLE ZYX
    BCC FINDS3    ;SIGN +
    CMP #$C0
    BCS FINDS2    ;SIGN +
    DEC COSXSIGN,X ;SIGN -
FINDS2
    CLC    ;ADD QUARTER PERIOD
FINDS3  ADC #$40 ;TO COS.
    AND #$7F ;MODULO HALF
        PERIOD

    TAY
    LDA SINE,Y
    STA COSX,X
;-----
;--- SINUS ZYX
;-----
    LDA ANGX,X ;ANG IS ALSO SIGN
    AND #$7F    ;LIKE BEFORE
    TAY
    LDA SINE,Y
    STA SINX,X
;---
    DEX
    BPL FINDSINE
```

;-----
INX – **czyli LDX #\$00** – ale po co, jak tu jeden bajt zamiast dwóch – jesteśmy w RAM stacji tu jest jej mało (2KB włącznie ze stosem i stroną zerową)

```
;-----
ROTPOINTS
    TXA
    STA LICZEW

    ASL A
    ADC LICZEW
    TAX    ;POINTNR *3
    LDY #$00
```

Pozycje początkowe wierzchołka

```
LDA BRYLA,X
STA XPRIM    ;XPRIM=X
```

```
LDA BRYLA+1,X
STA YPOINT
LDA BRYLA+2,X
STA ZPOINT
```

```
;-----
;- Y PRIM
;-----
    ; LDX ANGLEX
    ; BNE AXISX
    ; LDX YPOINT
    ; STX YBIS
    ; JMP STOREZPRIM
```

Obrót wokół OSI X – zarezerwowane linie, to sprawdzenie czy można pominąć obrót wokół X jeżeli kąt jest zerowy – można odznaczyć i używać...

```
AXISX
    LDY SINX
        ;IN ACC ZPOINT
    EOR ANGX
    STA SIGN
    LDA ZPOINT    ;SIN(ANGX)*Z
    JSR PROCKA
    STA DANA1
```

```
    LDY COSX
    LDA YPOINT    ;COS(ANGX)*Y
    EOR COSXSIGN
```

```
    STA SIGN
    LDA YPOINT
```

```
    JSR PROCKA
    SEC    ;COS(X)*Y - SIN(X)*Z
    SBC DANA1 ;SUBSTR.
```

```
    STA YBIS    ;Y PRIM = Y BIS
```

```
;-----
;--- Z PRIM
;-----
```

```
    LDY SINX
    LDA YPOINT
```

```
    EOR ANGX
    STA SIGN
    LDA YPOINT    ;SIN(ANGX)*Y
    JSR PROCKA
    STA DANA1
```

```
    LDY COSX
    LDA ZPOINT    ;COS(ANGX)*Z
    EOR COSXSIGN
```

```
    STA SIGN
    LDA ZPOINT
```

```
    JSR PROCKA
    CLC    ;SIN(X)*Y + COS(X)*Z
    ADC DANA1 ;SUM
    STOREZPRIM
```

```
    STA ZPRIM
;-----
;--- X BIS
;-----
    ; LDX ANGLEY
    ; BNE AXISY
    ; LDX XPRIM
    ; STX XBIS
    ; JMP STOREZLAST

Obrót wokół OSI Y
AXISY
    LDY COSY
    LDA XPRIM
    EOR COSYSIGN
    STA SIGN
    LDA XPRIM    ;COS(ANGY)*X'
    JSR PROCKA
    STA DANA1

    LDY SINY
    LDA ZPRIM    ;SIN(ANGY)*Z'
    EOR ANGY
    STA SIGN
    LDA ZPRIM

    JSR PROCKA

        ;COS(Y)*X' + SIN(Y)*Z'
    CLC
    ADC DANA1 ;SUM

    STA XBIS
;-----
;--- Z BIS = Z LAST
;-----

    LDY SINY
    LDA XPRIM
    EOR ANGY
    STA SIGN
    LDA XPRIM    ;SIN(ANGY)*X'
    JSR PROCKA
    STA DANA1

    LDY COSY
    LDA ZPRIM    ;COS(ANGY)*Z'
    EOR COSYSIGN

    STA SIGN
    LDA ZPRIM

    JSR PROCKA
    SEC
        ;SIN(Y)*X' - COS(Y)*Z'

    SBC DANA1 ;SUBSTR.

    STOREZLAST
        ;!!!!RULE FOR Z OBSERV.
    CLC
    ADC #$80
    TAY
    LDA ZDIV,Y
```

STA ZLAST ;ZBIS = ZLAST

Czary-mary dla perspektywy

;-----

;--- X LAST

;-----

; LDA YBIS

; LDX ANGLEZ

; BNE AXISZ

; LDX XBIS

; STX XLAST

; JMP GETYLAST

Obrót wokół OSI Z

AXISZ

LDY SINZ

LDA YBIS

EOR ANGZ

STA SIGN

LDA YBIS ;SIN(ANGZ)*Y"

JSR PROCKA

STA DANA1

LDY COSZ

LDA XBIS ;COS(ANGZ)*X"

EOR COSZSIGN

STA SIGN

LDA XBIS

JSR PROCKA

SEC

;SIN(Z)*Y" - COS(Z)*X"

SBC DANA1 ;SUBSTR.

STA XLAST

;-----

;--- Y LAST

;-----

LDY SINZ

LDA XBIS

EOR ANGZ

STA SIGN

LDA XBIS ;SIN(ANGZ)*X

JSR PROCKA

STA DANA1

LDY COSZ

LDA YBIS ;COS(ANGZ)*Y

EOR COSZSIGN

STA SIGN

LDA YBIS

JSR PROCKA

CLC ;SIN(Z)*X + COS(Z)*Y

ADC DANA1 ;SUM

GETYLAST

;TAY ;YLAST

Perspektywa X2D

;-----

PRSPCT

;---

LDY ZLAST

EOR ZLAST ;LOSE ACC

STA SIGN

EOR ZLAST ;RECALL ACC

JSR PROCKA

LDY ZOOM

ZOOM X2D

CPY #\$FC ;ZOOM FACTOR=1?

Czy trzeba mnożyć czy prescale = 1?

BCS STOREY2D

STA SIGN

TAX ;ONLY FOR

NEGATIVE INDICATOR

JSR PROCKA

STOREY2D

LDX LICZEW

STA TABY2D,X

;-----

Perspektywa Y2D

LDY ZLAST

LDA XLAST

EOR ZLAST ;LOSE ACC

STA SIGN

EOR ZLAST ;RECALL ACC

PERSP4

JSR PROCKA

LDY ZOOM

CPY #\$FC

BCS STOREX2D

STA SIGN

TAX

JSR PROCKA

STOREX2D

LDX LICZEW

STA TABX2D,X

CNTROT

Wierzchołek obliczony – są następne?

INX

CPX ILEW

BEQ RTPSEX

JMP ROTPOINTS

RTPSEX

RTS

;-----

SINE

.BYTE \$00,\$06,\$0C,\$12,\$19,\$1F

[...]

PONAD PLAN

Ponad plan pokazałem Wam szybką perspektywę, jednakże uznałem, że warto pokazać jakieś inne ścieżki do własnych poszukiwań. Także „na górkę” dorabiam wersję z możliwością CYBER VECTOR, żeby pokazać, jak niewiele wystarczy do kompletnej zmiany efektu. Nie zmienia się nic w algorytmie, tylko sposób reprezentacji grafiki...

CZYTAJCIE!

Istnieją również inne sposoby na wyliczanie wektorówek. Można dokonywać obliczeń w układzie ortogonalnym i operować na macierzach. Tu polecam lekturę C=hacking w szczególności numerów 8,9,10,16,21 (www.

ffd2.com). Tu znajdziecie również seriously fast multiplication – super szybki sposób na mnożenie poprzez tablice kwadratów wymyślony przez George’a Taylor’a. Kapitalne artykuły „A Different Perspective” Część I do III.

Ciekawą metodę do grafiki 2D pokazał WAVEFORM w DISCOVERY issue #2 (również www.ffd2.com).

Wartościową literaturą jest książka „Grafika Komputerowa” R. Baumańa.

W książce „Assembler 6502” Ruszczyca, znajdziecie sposób na dzielenie liczby 16-bitowej przez 8-bitową. Znajdziecie go także na codebase, jak również ciekawe mnożenie liczb 8-bitowych, które jeżeli liczby są małe działa szybciej od naszego, a przy większych liczbach niestety wolniej.

„Programowanie grafiki wektorowej” C&A 95/08 strona 13 artykuł Przemka Cieślaka.

KOŃCOWY STUFF

Na koniec dziękuję za uwagę :-)

Wegi

Współczesne środowisko programowania na Commodore 64 (crossplatforming)

Tworzenie programów na C64 wydaje się sztuką niełatwą. Mała ilość dostępnej pamięci, ograniczone możliwości sprzętowe oraz „nieokienkowy” interface skutecznie zniechęcają potencjalnych twórców.

Kiedyś sprawa była dużo prostsza – w czasach gdy nie było Windowsów oraz internetu (a zapewniam, że były takie czasy), programista komodorowski dostawał do „ręki” **Turbo Assembler** i już był w siódmym niebie.

Był to edytor, który naprawdę dało się lubić – przemysłana konstrukcja, szybkość działania, ergonomia i chociaż pracował w trybie tekstowym „jednokienkowym” to na tamte czasy trudno było nawet sobie wyobrazić coś lepszego.

Ale mamy już XXI wiek i przez te kilkanaście (kilkadziesiąt) lat i na polu środowiska dla programisty powstały nowe wspaniałe rozwiązania, których komcio z racji swojego wieku nie doczekał. Zaraz, zaraz, czy aby na pewno?

WYRZUTY MORALNE

Wielu z obecnych programistów C64 nadal korzysta z „TurboAssa”, czy to pod emulatorem, czy na prawdziwym sprzęcie. Używają, bo uważają go po prostu za najlepszy i już.

A ja znowu uważam, że tak nie jest, a powody tego stanu rzeczy mogą być zupełnie inne. Może to strach przed nowymi metodologiami, może szpan, a może zwykłe lenistwo przed przełamywaniem barier tłumaczone przyzwyczajeniem. Oczywiście zaraz podniosą się protesty na temat ideologii retro, która nierozdzielnie wiąże się z samym C64 – a pewnie znajdzie się też szereg powodów, dla których warto dalej męczyć leciwego Turboassa. Wiadomo, klimat retro związany z C64 to właśnie to, co utrzy-

muje tą starą konstrukcję nadal przy życiu. Ale jeśli chodzi o przytaczany **TurboAssembler** to jest to już tylko narzędzie, a nawet jedno z wielu podobnych na C64 – „środek” za pomocą którego, mamy wyciskać z c64 rzeczy nieprzeciętne, a nawet nieprawdopodobne. Są lepsze narzędzia, a jak to mówią – „cel uświęca środki”;-)

Artykuł ten, oczywiście nie ma na celu obrażanie obecnych użytkowników Turboassa, a zachęcenie ich do używania współczesnych narzędzi dostępnych na komputerach PC, pomocnych przy produkcji oprogramowania, niezrządkiem dedykowanych samemu c64, a innych do rozpoczęcia przygody ze światem programowania. Programowanie wbrew pozorom, nie jest wcale, aż takie trudne, na dodatek samo wciąg.

A co dzisiaj mamy na „rynku”? Może zaczniemy od tego co powinno wchodzić w skład środowiska programisty.

EDYTOR

Jeśli chodzi o edytory, to specjalnie dedykowany dla C64 znam tylko jeden. Inna sprawa, że przy wyborze edytora nie kierujemy się tak mocno docelową maszyną, a wygodą edycji tekstu (plików tekstowych), a tych na sieci jest masa, po odpowiedniej konfiguracji pasuje każdy. **Relaunch64**, to propozycja Daniela Ludecke (<http://www.popelganda.de/>), który wykonał wersję edytora tekstowego dedykowaną właśnie dla Commodore 64. Już sam fakt, że to produkt na PC dla komodorowców, wręcz zobowiązuje do chociaż krótkiego przyjrzenia

```

dj1      lda y0
         cmp #200-8
         bcc dj2
         lda $d012
         sta hh0+1
         and #200011111
         clc
         adc #$40
         sta hh0+2
         rts

dj2      lda y0
         clc
         adc #5
         sta y0

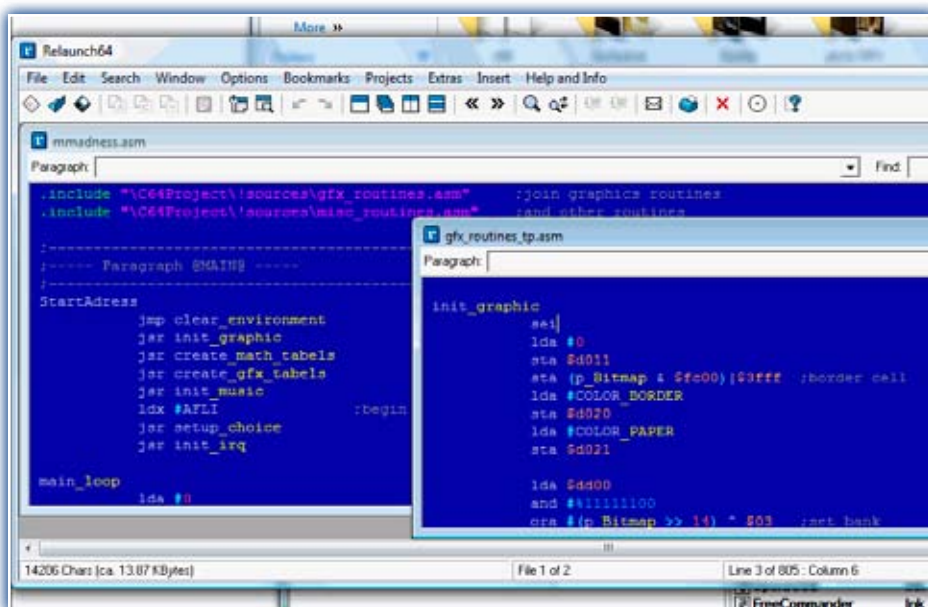
         lda y0
         lsr a
         lsr a
         lsr a
         tax

turbo-ass v4.1 - updated by tycoon/prime
x:15 line:325 bot:8a9b insert: line
  
```

Staruszek TurboAssembler

się tej pozycji. Używam go osobiście i uważam, że w ogólnej ocenie nie jest on taki ostatni. Znajdziemy w nim wszystko co jest niezbędne do komfortowej edycji tekstu, również przy pracy na wielu plikach. Menu są układane wg znanych windowsowych konwencji i nie trudno jest znaleźć to czego szukamy. Widać, że autor też zaczynał od **TurboAssa** bo są też „ściągnięte” dobre rozwiązania ze staruszka – np. bookmarki – czyli znaczniki zdefiniowane przez użytkownika do szybkiego przemieszczania się w ważniejsze części kodu poprzez skróty klawiszowe, czy skok do wybranej linii. Akurat do przemieszczania się po pliku, dodał również ciekawą opcję w postaci tzw. Paragrafów. Są to dodatkowe etykiety, które program przy otwarciu pliku automatycznie wyszukuje i można wtedy natychmiast się do nich dostać z listy rozwijanej, która jest umieszczona zawsze na górze okna – są one oprawione w średniki, które dla kompilatora (programu zamieniającego tekstową postać algorytmu na binarną gotową do uruchomienia na c64) stają się zwykłymi komentarzami i nie przeszkadzają w analizie kodu. Nie będę tu wspominał o opcjach wyszukiwania, zamiany, dostosowywania wyświetlanego tekstu, układzie okienek i innych tego typu oczywistych sprawach, tylko zapewnię, że temat ten dla tego edytora jest stuprocentowo wyczerpany. Zwykle kopiuj, wklej z innych źródeł t.j. codebase64.org lub innych dokumentów, czy nawet z okna GaduGadu kolegi kodera teraz stało się po prostu normą, a nigdy nie będzie to osiągalne dla pocziwego **TurboAssa**.

Relaunch64 – jak przystało na narzędzie dla programistów – posiada interfejs dla kompilatora, bo dopiero kompilator zapewni nam, spojrzenie na owoc naszej pracy (czyt. program). Edytor pozwala na „podpięcie” aż trzech różnych kompilatorów z oddzielnymi parametrami wywołującymi i dla każdego też, jest przypisany oddzielny klawisz skrótu. O ile kompilator utworzy nam plik wykonywalny, będziemy mogli wreszcie nasze dzieło uruchomić na emulatorze (lub prawdziwym C64) i sprawdzić wyniki. Edytor i tu również pozwala na konfigurację dla trzech różnych „emu”. O temacie kompilatorów i emulatorów wspomnę później, w dalszej części artykułu. A co jeszcze dla komodorowców w samym edytorze – sporo! Można dołączyć do naszego środowiska kompresor danych (np. exomizer), importer źródeł (np. z **Turbo Assemblera**), wstawiać bajty z plików, tworzyć tablice sinusów, konwertować wartości. Można definiować własne szablony do wstawiania, nawet dla zapominalskich są krótkie „gotowce” odnośnie tematu C64 - rejestry VIC-a, SID-a, CIA z przykładami jak np. obsługiwać klawisze, wyświetlić obrazek we FLI itd. Są również zaimplementowane funkcje do zarządzania plikami z kodem, podział na projekty – jednym słowem „profeska”. Oczywiście



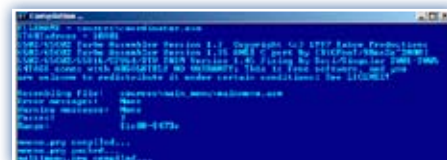
Tak się teraz pisze programy na C64 ;-) - Relaunch64 i jego podwórkę.

ście zapraszam do samodzielnego odkrywania opcji Realuncha, bo nie sposób ich tu wszystkich wymienić.

KOMPILATOR (CROSSASSEMBLER)

W tym temacie dla C64 możemy naprawdę przeprobować: **DreamAss**, **Acme**, **Kick Assembler**, **Asm64**, **64Tass** i znajdzie się pewnie wiele innych. Jako, że jak wcześniej wspomniałem, chciałbym namówić użytkowników Tasma do przesiadki na nową technologię to proponuję ten ostatni **64Tass**, pierwotnie napisany przez Polaków z bardzo znanej niegdyś grupy TA-BOO. Dlaczego akurat taki wybór? Bo... jego składnia w STU procentach pokrywa się z **Turbo Assemblerem**, plus oczywiście dodatkowe funkcje. A więc nie trzeba się uczyć korzystania z assemblera na nowo, ale od razu używać i tworzyć.

Co nam daje korzystanie z takiego rozwiązania? Oprócz, niewątpliwie wygody edycyjnej (dochodzi również myszka), to mamy nieograniczony dostęp do całutkiej pamięci komodoraka bez wyjątku – dla **Turbo Assemblera** zawsze to było jakieś ograniczenie, co prawda powstały wersje na rozszerzone pamięci, ale zawsze coś zajmowały. Druga rzecz, to nielimitowana długość kodu, bo nawet wersja Tasma na dodatkowy RAM, miała ograniczenie w postaci maksymalnej liczby linii kodu do 4096 - dużo? Oczywiście zależy od projektu, ja potrafię w swoich produkcjach zwiększyć tę liczbę bez problemu 10-cio krotnie. Trzecia rzecz – to komentarze, nareszcie bez ograniczeń, i dużo, i jakie się chce, i nie zajmują miejsca w komcie. Czwarta pomocna cecha to długość etykiet – nareszcie można cieszyć się długimi nazwami np. Tabela_sinusa_dla_wartosci_x_warstwy_sprites, a nie jak poprzednio tsdwxsl – bo tylko na 8 znaków w etykiecie pozwalał **Tasm**. Chyba nie muszę dodawać jak te dwie ostatnie



Kompilator w akcji

cechy wpływają na przejrzystość kodu i nawet spojrzenie na niego 10 lat później nie kończy się zagadką dla Sherlocka Holmesa. Całości dopełnia to, że nie ciśniemy się w 40 znakach na ekranie, a w tłu pozwala nam pulpit w Windowsie (a przecież ostatnio królują panoramiczne monitory).

Teraz również, nic nie stoi na przeszkodzie, aby dzielić kod na oddzielne pliki (moduły) dla przejrzystości (np. procedury graficzne oddzielnie, dla obsługi przerwań oddzielnie), oraz doklejać dane dla grafiki i muzyki. Wystarczy tylko scalać te dane za pomocą dyrektyw `.include` `.binary` w dowolnym miejscu i momencie naszego pliku z kodem.

Nie będę opisywał tu wszystkich cech kompilatorów bo to temat rzeka, manuale są ogólnie dostępne, składnia jest we wszystkich podobna, a użyteczność indywidualnych funkcji to już subiektywna ocena przyszłego użytkownika. Jeszcze tylko wspomnę, że każdy posiada dostęp do tzw. meta-instrukcji – którymi możemy kształtować dane jakie pojawiają się w pamięci komcia po kompilacji (składnia zapożyczona głównie z języka C), ale używanie ich oczywiście nie jest obowiązkowe. Kompilatory, same w sobie są programami uruchamianymi się w oknie DOS-u, tam też wyświetlają informacje o postępie kompilacji, ale również o napotkanych błędach. Przy odrobinie dobrej woli, można skonfigurować sobie środowisko tak, aby błędy naszego programu wyświetlały się w edytorze Relaunch64 – a zaleta tego

rozwiązania jest chyba oczywista. Sam proces kompilacji trwa w mgnieniu oka, a to ważne bo uruchamiać poprawiany kod możemy i 1000 razy dziennie. Program z naszymi danymi i kodem, wraz z uruchomieniem na emu jest już na swoim miejscu i nie trzeba czekać na jakieś wgrywanie albo przemieszczanie danych.

Widać, że ten sposób tworzenia jest dużo bardziej wydajny niż ślęczenie przed Turbo Asem, nie martwimy się błędami „zawieszającymi” komcia i nic już teraz nie stoi na przeszkodzie, aby porywać się na coraz to większe i ciekawsze projekty.

EMULATOR

Aby rozpocząć programowanie na „crossplattwormie” został nam do wyboru jeszcze tylko emulator, ale w tym temacie, myślę że każdy ma sporo doświadczenia, jak i również swojego faworyta. Ja do programowania używam Vice, jest szybki, posiada całkiem rozsądny monitor do podglądu tego „co się dzieje na ekranie”, ale jeśli chodzi o poprawność emulacji bywają lepsze (przynajmniej na chwilę obecną). Ma on jednak niewątpliwie jedną główną dla mnie zaletę – uruchamia się zawsze niezależnie od konfiguracji PC-ta.

Zdradzę, że całe środowisko do programowania na c64 trzymam i uruchamiam na tzw. pendrive USB, w ten sposób mogę zająć się programowaniem na komci na każdym dostępnym komputerze PC w polu widzenia :) Niestety na moim służbowym laptopie (a nadmienię, że nie pracuję jako informatyk), mam sporo ograniczeń założonych przez administratorów i właśnie Vice jest jednym z niewielu emulatorów które w ogóle działają – i to bezproblemowo. Emulator dla programisty to coś więcej niż wierne odtwarzanie naszego ulubionego komodoraka, to również narzędzie do podglądu zawartości pamięci i rejestrów, wstawiania tzw. breakpointów, a więc kontrolowania poprawności działania naszego programu. Niewątpliwą zaletą jest możliwość przyspieszania emulacji (tryb warp), ale również zwalniania jej

(np. do jednej klatki na sekundę – ważne przy docyfkowywaniu efektów), opcje te nierzadko oszczędzają masę czasu programisty, czyniąc pracę nad projektem niesamowicie skuteczną, czyli wydajną i bardziej dopracowaną, ale także wygodną.

NARZĘDZIA (TOOLS)

Temat kodowania na PC, ale dla C64, mamy w sumie załatwiony, ale to nie wszystko. Istnieje bowiem dostęp do całej masy programów narzędziowych wykonanych głównie przez ludzi ze sceny, z myślą tylko o wykorzystaniu ich dla celów C64. Konwertery grafiki, kompresory danych, edytory graficzne lub muzyczne, kreatory dyskietek, itd.

Nie można też narzekać na bazę wiedzy na tematy związane z kodowaniem na c64, która znajduje się w dowolnym momencie poprzez przeglądarkę WWW w internecie. Stron jest setki, a takie jak np. codebase64.org dodają tylko smaczku, i uświadamiają że temat kodowania na c64 to nie jakiś niszowy wymysł wariata, a „bawi” się w to mnóstwo osób na całym świecie. Faktem jest, że zdecydowana większość materiału dostępna jest w języku angielskim, ale może kiedyś doczekamy się podobnej pozycji w naszym języku.

DO ROBOTY!

Na koniec mała niespodzianka. Wykonałem taki przykład środowiska programisty gotowego do wykorzystania! Jedyne co trzeba zrobić to ściągnąć plik C64Project.zip z działu DOWNLOAD ze strony redakcji ca-fan.pl i rozpakować w głównym katalogu na którejś z partycji na własnym dysku twardym (lub pendrive). Powinien wtedy pojawić się w katalogu C64Project, w którym znajdziemy kolejne katalogi. Na początku wejźmy w „editor” i możemy uruchomić naszego Relaunch-a. Wiercie albo nie, ale już od tej chwili można zaczynać programować w assemblerze i uruchamiać (klawisz F6) własne programy. Oczywiście nie zabrakło również przykładowego programu, który możemy znaleźć w katalogu !/sources.

Otwieramy go z menu Open poprzez plik „przyklad.asm” i możemy również od razu go uruchomić (F6).

Programik wyświetla obrazek wykonany konwerterem MUSCU Hires, z dodatkowym naszym napisem oraz odtwarzający muzyczną w tle – taka prezentacja, może mieć zastosowanie np. jako własna strona tytułowa. Klawisz spacji kończy prezentację, a potem możemy dodawać swój własny kod.

Plik obrazka można oczywiście podmienić, jest on w katalogu !sources/bin/obrazek.prg. Tak samo możemy postąpić z plikiem do muzyki !sources/bin/muzyczka.prg. Można również zmienić treść wyświetlanego tekstu – tekst zmieniamy w cudzysłowach zmiennej wyświetlany_text. Trzeba tylko oczywiście pamiętać, że przed uruchomieniem (F6) trzeba zapisać zmiany(ctrl+s).

Środowisko jest na tyle skonfigurowane, aby można było bezproblemowo zacząć pisać i uruchamiać własne programy, a pliki wynikowe po poprawnej kompilacji powinny się pojawiać w katalogu głównym C64Project.

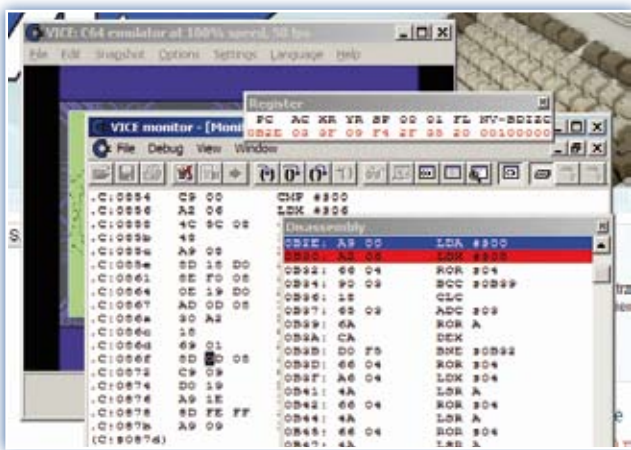
Katalog ten zawiera jeszcze inne katalogi poza /editor i !/sources. Można znaleźć w nim /emulator, w którym znajduje się zubożona wersja WinVice 2.2 - w zupełności wystarcza ona do uruchamiania naszych prac. Kolejnym katalogiem jest /help w którym umieściłem manuały i helpy odnoszące się do kompilatora, i innych narzędzi, oraz różne „pomocniki” przy kodowaniu, zachęcam do przejrzenia tego katalogu na początku, może rozjaśni wiele wątpliwości. Kolejny to /log który zawiera logi z procesu kompilacji (błędy są wyświetlane w oknie edytora), oraz plik z listą etykiet użytych w naszym programie, wraz ich fizycznymi adresami w pamięci C64 - przydatna rzecz przy debugowaniu. W ostatnim /tools znajduje się kilka narzędzi pomocnych przy tworzeniu własnych produkcji. Jest tam też główny plik kompilatora 64tass.exe, poprzedzony plikiem 64tass.bat, który właśnie jest uruchamiany podczas polecenia kompilacji - modyfikuje lekko środowisko dla większej wygody (temat raczej na inny artykuł).

Potem mamy klika innych ciekawych plików narzędzi, ale właśnie w zależności od zainteresowania tym tematem, może będziemy im się przyglądać w kolejnych artykułach, a kto wie, może uda się stworzyć jakiś mały kurs programowania. Na tę chwilę wyróżnię tylko plik „MUCSU-Hires Converter.exe”, dzięki któremu możemy stworzyć własne obrazki dla podanego wcześniej przykładowego programu.

Na tym kończę ten wywód, mam nadzieję, że tak podane „na tacy” rozwiązanie, zachęci chociaż część z was do spróbowania swych sił w crossplattwormingu, w końcu niech się przydadzą do czegoś te PC-ty;-)

Mam również nadzieję, że dzięki temu tematowi, przygodę z programowaniem na C64 rozpoczną także nowe osoby. Czekam oczywiście na wasze reakcje, opinie i spostrzeżenia.

Skull/Samar



WinVice i jego monitor

HANDLARZE GIEŁDOWI

W połowie lat 80-tych w naszym kraju zaczęły powstawać pierwsze giełdy komputerowe, a razem z nimi przybywali ludzie, którzy chcieli na tych giełdach zarobić. Bez ludzi handlujących komputerami, osprzętem do nich, a nawet produkcjami – nie istniałyby takie miejsca. Tam spotykała się młodzież głodna wiedzy komputerowej i poszukująca oprogramowania do swojego komputera. Ostatnio trochę pisze się o giełdach komputerowych i wspomina czasy tam spędzone.

Jakiś czas temu na Internecie, po publikacji wywiadu V-12 z Waldemarem Czajkowskim, na różnych forach internetowych (polskich i zagranicznych) rozpoczęła się dyskusja. Tych, którzy nie czytali jeszcze wspomnianego wywiadu zapraszam na stronę: http://www.riversedge.pl/wywiad_z_waldemarem_czajkowskim

Waldemar Czajkowski w od połowy lat 80-tych do połowy 90-tych był jednym z największych i najprężniej działających w Polsce handlarzy pirackim oprogramowaniem – aż do momentu wejścia w życie ustawy o prawach au-

torskich. Jeśli ktoś miał głowę na karku i trochę smykałki do interesu, to w tym czasie mógł zrobić na pirackich grach sporo kasy i nie były to dodatkowe drobne do kieszonkowego. Nie trzeba było dużych inwestycji w taki biznes, poza tym nie musiało się płacić twórcom gier odpowiednich tantiem. Tacy ludzie szybko się bogacili, bo rynek chłonął oprogramowanie (głównie różne gry), szczególnie dla Commodore 64 i Amigi. To właśnie dzięki tej masie programów użytkowych i gier można było na tym zarobić.

Nasuwa się proste pytanie: czy warto ludzi, którzy sprzedawali pirackie gry uważać za coś wyjątkowego i jakoś ich wywyższać? Dla większości ludzi z tamtego okresu okazują się oni zwykłymi złodziejami, którzy po prostu na cudzym soficie zarabiali krocie, a zwykli programiści w naszym kraju, sprzedający legalnie, nie mogli się utrzymać. Z perspektywy czasu inaczej można ich oceniać, zresztą – w tamtych czasach odradzało się nasze państwo i na pewne rzeczy Polska jeszcze nie była przygotowana (jak choćby wspomniana ustawa o prawie autorskim).

Przekonajmy się, co sądzą inni miłośnicy komputerów Commodore/Amiga, którzy dorastali w tym okresie. Oto wypowiedź Pampam'a: (...) Ja mam trochę niepochlebne zdanie o takich koleśiach z tego względu, że oni na takich, którzy coś potrafili, też nieźle jechali. Weźmy taki przykład: Ultimate Puke zrobił koleśowi cardridge (tj. jakiś tam gotowy zestaw softów, które handlarz chciał). Nikt jeszcze wtedy nie potrafił takich rzeczy, przynajmniej spośród tych osób, które znałem i handlarze raczej kopiowali gotowce... Kasa, którą za to dostał było śmieszna w stosunku do tego, ile handlarz na tym zarobił, więc... robienie w tej chwili bohaterów z takich koleśi dlatego, że kopiowali hurtowe ilości kaset czy pirackich kardridży, jest głupie... Owszem, można zrobić wywiady, ale na pewno bez gloryfikowania w nich tego typu postaci – raczej ze sceną to oni nie mieli wiele wspólnego. My często staraliśmy się na początku rozprawdzać przez takich ludzi dema, dając im je za darmo. Ale z tego, co wiem, to często mieli to w „gdzieś” i sprzedawali tylko gry, bo to schodziło i na tym robili kasę. A na kasie im tutaj tylko zależało... Jak wiemy, scena to raczej duch, a nie robienie pieniędzy.

Jednak tacy ludzie mieli swojego zasługi w komputeryzacji kraju, więc to jakiś kawał historii – czy dobrej, czy złej, ale wartiej odnotowania i przekazania następnym pokoleniom. Nie są oni jakimiś wielkimi miłośnikami naszych komputerów. Gdyby można było zarobić na innym komputerze, to by się na niego przerzucili i na nim robili dalej interes. Wykorzystywali lukę w prawie polskim i nic ich nie obchodziło prócz zysku. Trzeba odróżnić tzw. płatki (ludzi, którzy sprzedawali parę gier na giełdzie, gdzie dorabiali sobie do kieszonkowego) od takich, którzy robili to hurtowo i z tego żyli. I trzeba dodać, że zazwyczaj dobrze sobie żyli i dorabiali się sporej ilości pieniędzy. Tak tamte czasy wspomina znany scenowiec Odyn: To były kosmiczne czasy, mało kto miał jakkolwiek dostęp do BBS'ów, nie mówiąc już o posiadaniu własnego modemu. Jeżeli nie swapował na flofach, zostawała mu tylko giełda,



więc osoby spoza demosceny były wręcz na nią skazane. Dobrze pamiętam giełdę w Bielsku, gdzie sam dorabiałem kopiując z Agothem soft na C64, ale były to sporadyczne wypady z dwiema stacjami i burstem. Wiadomo, nie było wtedy żadnego prawa autorskiego. Więc imho w 90% przypadków giełda była jedynym źródłem komercyjnego softu, jak również często miejscem spotkań braci scenowej. Z punktu widzenia ówczesnego zaoferowania naszego kraju i braku kanałów dystrybucji oprogramowania z prawdziwego zdarzenia, giełdy komputerowe w latach 90-tych były pozytywnym zjawiskiem. Przecież składanki softu na kasetach czy flokach w takiej samej formie można było nabyć w sklepach z kompami 8-bit czy Amigami. Wtedy zarówno sklepy, jak i giełdy sprzedawały ten sam stuff, przy czym pamiętam, że na giełdzie bardziej dbano o klienta. Kiedyś kupiłem jakiegoś zwalonego freeza na kasiecie i za tydzień na giełdzie wymieniono mi go na inny 'zestaw'. Więc moim zdaniem giełdy w Polsce, w tamtych dzikich czasach raczkującego kapitalizmu, były czymś normalnym, jakimś pierwszym ogniwem dystrybucji softu w ogóle. Jedyne, czego oczywiście nie pochwalałem, to sprzedawanie scenowego warezu (dem, magazynów) za kasę, co często wytykałem ówczesnym „biznesmenom”.

Najgorsze w tym procederze było to, że ci ludzie sprzedawali też produkcje typowo scenowe, kiedy – jak wiadomo – są one za darmo. W pierwszych produkcjach demosceny można spotkać teksty, że dany program jest public domain i nie wolno go sprzedawać. Niestety, zazwyczaj handlarze giełdowi mieli to „gdzieś”. Z drugiej strony, gdyby nie oni, to zapewne wiele gier i programów użytkowych nie zachowałoby się do dnia dzisiejszego. To dzięki tym ludziom rozchodziły się mało znane polskie produkcje – w szczególności gry czy programy użytkowe.

Trzeba powiedzieć otwarcie, że większość osób ze sceny właśnie tak zaczynała swoją przygodę z komputerem: od sprzedawania gier na giełdzie komputerowej. Znane postacie z demosceny handlowały i robiły na tym niezły interes, choć może nie na taką skalę, jak Waldemar Czajkowski. To właśnie giełdy komputerowe były miejscem, gdzie z handlarzy przeistaczali się w ludzi demosceny. To było miejsce, gdzie ludzie zaopatrywali się w różnego rodzaju produkcje. Jak ciekawie powiedział Pampam: (...) można powiedzieć żartobliwie: tak, jak katolicy chodzą do kościoła co niedzielę, tak właściciele i fani komputerów zjeżdżali z okolic, żeby pojawić się na giełdzie, pooddychać tym powietrzem, popatrzeć na różne kopmy itp.

Wiadomo, żeby gdyby nie giełda i handlarze, to demoscena trochę wolniej by się rozwijała, podobnie jak i cała informatyka w naszym

kraju. To handlarze dostarczali nam brakujący sprzęt czy odpowiednie gry, a czasami różne programy użytkowe.

Oto co sądzi o tym twórca Black Box'a, pan Romuald Drahokaupil: Oczywiście że tak, jeśli wywiad czy artykuł jest ciekawy i wnosi coś nowego do tego tematu. Na różnych forach internetowych internauci mogą wyrażać rozmaite poglądy na ten temat, ale ja mam mieszane uczucia w kwestii takich polemik. W tego rodzaju wypowiedziach rzadko gości obiektywizm. Zwykle są to jednostronne osądy będące odzwierciedleniem poglądów uczestników tych forów. W takich dyskusjach czy prędkiej, czy później dochodzi do polaryzacji stanowisk, co niemal zawsze prowadzi do skrajności wypowiedzi. Ewidentnym tego przykładem może być sprawa Romana Polańskiego. Jedni odsądzają go od czci i wiary, uważając, że jako gwałcieł dziecka popełnił czyn tak wysoce haniebny, że nawet próby usprawiedliwiania go są rzeczą haniebną, a inni uważają go za ofiarę młodocianej prostytutki, którą podstawiała mu jej matka, licząc na uzyskanie jakichś korzyści dla siebie i swojej córki. Mało kogo stać tu na bezstronność. Ludzie bowiem niemal w każdej sytuacji chcą wyrażać swe ekstremalne opinie; chcą być za albo przeciw. Zupełnie nie przejmują się moralnym imperatywem, który zaleca, aby nie osądzać.

Istotnym pytaniem jest więc nie to, czy pisać, bo pisać można o wszystkim i w wszystkich, ale to, jak pisać. A temat handlowania na giełdach to temat-rzeka i można by o tym rozpisywać się bez końca.



Rozpatrując sprawę w aspekcie historycznym należałoby zróżnicować rolę handlujących na giełdzie w czasach rządów komunistów i w czasach współczesnych. Co innego było wówczas dopuszczalne, a co innego jest teraz. Inna była też wówczas rola społeczna takich ludzi, a inna jest obecnie. W tamtych czasach dostarczali oni bowiem tego, czego nie można było dostać w słabo zaopatrzonych państwowych sklepach. Szczególną rolę odegrali w rozpowszechnianiu wiedzy o komputerach domowych, bowiem kopiując i sprzedając oprogramowanie do tych komputerów, przeważnie zachodniego pocho-

dzenia, przyczyniali się do rozwoju informatyki w kraju, który pozostając przez długi czas „za żelazną kurtyną”, był w tej dziedzinie bardzo zapóźniony. Trzeba podkreślić, że ówczesne prawo karne nie uważało tego za przestępstwo i dlatego takie działania nie były ścigane.

Czy warto ludzi, którzy sprzedawali pirackie gry uważać za coś wyjątkowego i jakoś ich wywyższać? No cóż, odpowiadając na pytanie zawarte w tekście uważam, że nie ma absolutnie żadnego powodu, aby ich w jakikolwiek sposób wywyższać czy też uważać za coś wyjątkowego (zaznaczam, że chodzi mi tu o hurtowników, a nie osoby dorabiające sobie na swoich zainteresowaniach.) Byli to po prostu zwykli handlarze, podobnie jak dziś osoby sprzedające auta czy buraki. Oni po prostu zajmowali się handlem i niczym więcej, a że przypadkiem na sofcie dało się sporo zarobić i nie wiązało się to z konsekwencjami prawnymi, to tym się właśnie zajęli. Większość z tych ludzi kompletnie nie interesowała się komputerami czy programowaniem. Obsługę sprzętu opanowali w stopniu koniecznym do zarabiania kasy. Gdyby nagle okazało się na drugi tydzień, że jeszcze lepszą kasę mogą zrobić na sprzedaży buraków czy klocków LEGO, natychmiast by się na to przeczucili. Trudno też przypisywać im jakąś szczególną rolę w rozwoju komputeryzacji kraju. Oni robili na tym biznes, a wzrost edukacji informatycznej był tylko nic a nic nie interesującym ich skutkiem ubocznym ich działalności. Ani go nie wspierali, ani nie utrudniali – po prostu to samo wyszło i nie można im żadnych udziałów przypisywać.

Naiwnością było też ze strony sceny dawanie im swojego stuffu do dystrybucji i to jeszcze darmowej. Wiadomo, że koleś nastawiony na kasę nie będzie za darmo dystrybuował czyichś produkcji, tylko będzie starał się wyciągnąć z tego kasę. Jednak nie można zgodzić się z twierdzeniem, że oszukiwali („nie-żle jechali”) oni ludzi sceny, płacąc im mało za usługi czy też pisany przez nich soft. Wina jest tu raczej po stronie ludzi ze sceny, że tak mało za swe tak duże umiejętności żądali. Na ile się dogadali, to na pewno handlarz tyle im zapłacił, a że potem robił na tym kokosy, to już jego zysk lub też ewentualna strata (gdyby na przykład nikt tego nie chciał). Było nie było, ale pretensji mieć nie można. Jakby scenowcy chcieli więcej, to handlarze musieliby tyle dać albo zrezygnować z ich usług.

Była to jakoś historia i nie ma co się nad tym tematem bardziej rozwodzić. Ludzie, którzy w jakiś sposób chcą się dorobić na cudzych produkcjach byli i będą. Gdzie da się z czegoś wycisnąć pieniądze, tam zawsze ktoś się znajdzie.

MrMat & Ramos

STEIN EIKESDAL

Wywiad ten miał być przeznaczony do numeru piątego, ale jakoś się przeciągnął i dopiero teraz może być zamieszczony w tym numerze. Mało kto tak naprawdę wie kto to jest Stein Eikesdal znany bardziej jako Stone Oakvalley, ale jak ktoś wspomni o SOASC= czy SOAMC= to już zaczyna coś świtać w głowie. To właśnie Stein Eikesdal jest odpowiedzialny za stworzenie i prowadzenie kilku archiwum związanych z muzyką C64 i Amigą. Oprócz tego stworzył kilka ciekawych projektów o których możecie tu przeczytać.

Na początek przedstaw się. Nie wszyscy tutaj wiedzą co robisz i czym się zajmujesz. Powiedz, gdzie pracujesz, jakie masz zainteresowania. Ogółem powiedz coś o sobie.

Witam. Nazywam się Stein Eikesdal, znany jestem także jako Stone Oakvalley. Urodziłem się w 1974 roku. Od niedawna jestem szczęśliwie żonaty z Cheery May, nie mamy dzieci, mieszkamy w małej miejscowości zwanej Nedre Vats należącej pod Vindafjord (ok. 8200 ludzi) w Norwegii. Żyje tutaj około 1800-2500 ludzi w wioskach Ovre/Medre Vats, a do najbliższego miasta zwanego Haugesund mam jakieś 40 km. Vats jest naprawdę spokojnym miejscem, w którym występuje mnóstwo zieleni i masywy górskie żywo przypominające te z „Władcy Pierścieni”. Czasami jest tutaj naprawdę nudno, więc każdy stara się być kreatywny, co ja zacząłem bardzo wcześnie. Zaczynałem od LEGO we wczesnych latach 80-tych, później grałem na maszynie Coleco- vision na miejscowej stacji benzynowej. Prawdopodobnie ta maszyna skierowała mnie w stronę świata komputeryzacji. Pierwszą moją grą, było „Looping” w latach 80-85 (nie pamiętam dokładnie, chociaż staram się sobie przypomnieć szczegóły mojego dzieciństwa). Przypuszczam, że była jeszcze jedna maszyna, która skierowała mnie w stronę komputerów, ale niestety nie mogę sobie przypomnieć jej nazwy ani pochodzenia.

W okolicach roku 1984 nastąpiła dla mnie era **Atarii 2600** (nadal ją mam), z grami „Pong/



Sports Collection”, „Tanks” i „Asteroids”... Jestem osobą interesującą się fikcją naukową (Sci-Fi), więc siłą rzeczy gra „Asteroids” była dla mnie nr. 1. Rekordy w grach? Nie, to nie dla mnie. Nigdy nie dbałem o nabijanie punktów, tylko o grafikę i rozrywkę. W 1988 kolega ze szkoły przedstawił mi komputer **Commodore 64** i od tego się zaczęło. Kupiłem używany **C64** z magnetofonem, grami „Winter Games”, „Sorcery” i mnóstwem kaset nagranych w systemie Turbo z grami, demami i oprogramowaniem. Do tego dochodziła książka „Podstawy Commodore 64” (Commodore 64 Basic), ukazująca ukryte dla normalnego użytkownika, siły drzemiące w tej maszynie. Gry były oczywiście powalające, ale programowanie i ujrzenie jak piksele są rysowane na ekranie, okazało się najlepiej zapamiętaną rzeczą. Balon C64, poruszający się na ekranie, był dla mnie inspiracją do dalszych eksperymentów. Piksele okazały się dla mnie, jak klocki LEGO.

Od tego czasu świat w którym żyłem, opierał się na rysowaniu pikseli na kartce papieru,

pisaniu programów w BASIC’u i nie zwracaniu uwagi na nauczycieli podczas zajęć szkolnych. Kolejną imponującą częścią mojego życia, było **oglądanie dem na Commodore 64** do 3-4 w nocy. Pragnąłem wtedy robić podobne rzeczy do tych z oglądanych dem, Wtedy byłbym tak samo sławny jak ci koderzy, trenerzy i crakerzy... Fanatycznie czytałem mnóstwo scrolli i słuchałem mnóstwo muzyki. Jednak było jedno demo, które okazało się najlepsze. Tym demem było „It’s drum time!” Lukullus’a.

Perkusja na C64? Do momentu obejrzenia tego dema, Kompletnie tego nie słyszałem. Zapragnąłem wtedy kupić sekwencer perkusji dla C64. Nie przypuszczałem, że to będzie takie wspaniałe, ale od tego momentu chciałem tworzyć muzykę, czego wcześniej nie robiłem na C64. Spowodowane to było brakiem oprogramowania muzycznego, a muzyka tworzona za pomocą BASIC’a nie była taka bogata jak tutaj.

W roku 1989 przeszedłem obok wystawy w sklepie z zabawkami zawierającej **Amigę 500**.

To był dla mnie cios, a pierwszą rzeczą którą widziałem był Workbench 1.3 odpalonym z dyskietki, którą przyniósł jakiś dzieciak. Na tej dyskietce był dodatkowo SoundTracker. Kiedy usłyszałem go w akcji, nakręciłem się tak, że musiałem mieć tą maszynę. Kupiłem wtedy kasetę „Amiga 500 Promo Video”, która trwała ok. 30 minut i obejrzałem ją około 200 razy. Kiedy skończyłem 14 lat, moja siostra wraz z mężem, dali mi opcję do wyboru, która w późniejszych czasach była osławiona jako szczególnie dziwny wybór dokonany przez młodego wtedy Stone’a Oakvalley’a. Musiałem dokonać wyboru pomiędzy tygodniowym pobyt w Disneylandzie, połączonym ze zwiedzeniem studia Universal Pictures, a zwykłą Amigą 500 z programem „Sonix”!

Skoro interesowałem się efektami specjalnymi i sposobami tworzenia filmów, oraz komputerami, musiałem podjąć jedną z najtrudniejszych decyzji w mojej historii. Zastanawiałe się nad dokonaniem wyboru przez kilka tygodni, póki nie zdecydowałem się na Amigę 500. Poparałem to następującym wywodem: podróż do Disneylandu na pewno jest wspaniałym przeżyciem, jednak tylko przez określoną chwilę, która mija wraz z powrotem do domu... natomiast posiadając Amigę 500, można tą niesamowitą chwilę rozciągnąć w nieskończoność, więc wybór okazał się dla mnie bardzo logiczny.



Około 1990 roku sprzedałem swojego komodorę i używałem samej Amigi 500 programując w Amiga Basic, grając w gry i oglądając dema. Około roku 1992-94 sprzedałem 500-tkę i kupiłem **Amigę 1200** z dyskiem twardym 120MB. Wtedy też zacząłem tęsknić za komodorą tak bardzo, że postanowiłem kupić następnego. Tym razem nabyłem go od kolegi, który w zestawie posiadał także stację 1541. Nadal mam ten sprzęt, który służy mi w projekcie SOASC=, jako maszyna posiadająca układ MOS6581R4 w wersji PAL.

Dużo komponowałem na programie SoundTracker, jednak więcej pracowałem z Protrackerem w wersji 1.1b. Na zlocie „The Gathering 1997” wystawiłem dwa dema nazwane

„PCPuke” i „PowerDeath” na Amidze 1200. Zdobyły miejsca 13-te i 19-te, wraz z naszym 3-cio miejscowym demem „Firestarter/Prodigy”, które wyglądało jak teledysk. To było wspaniałe.

W 1998 roku sprzedałem Amigę 1200 i przesiedłem się na **90MHz peceta**. Zacząłem wtedy komponować muzykę w programie „FastTracker 2”, który okazał się dla mnie lepszy niż programy z Amigi, z powodu braku ograniczenia do czterech kanałów. Około roku 2003-2006 „odkurzyłem” **Amigę 500** (nabyłem ją za darmo w 1999 roku), **C64 „mydelniczkę”** i **nową wersję C64**. W latach 2007-2009 kupiłem trochę sprzętu amigowego i komodorowskiego, włączając w to **Amigę 1200 z czytnikiem kart Flash i kartą turbo Blizzard**, którą wykorzystuję przy pracy nad projektem **SOAMC=**.



Po zakończeniu projektu **SOASC=** w 2006 roku, przeszedłem do tak samo szaleńczego nagrywania muzyki z Amigą w latach 2007-2009. Przez te wszystkie lata, kiedy nabywałem C64, naliczyć mogę ponad **dwanaście egzemplarzy** z różnymi chipami i w wersji PAL lub NTSC, do tego dochodzi **Atari 7800, NES i Atari 1024STFM**. Mam jeszcze w planach kupno takich maszyn, jak Colecovision, Amstrad i być może sprzęt Atari. Kiedy znajdę jakiś ciekawy sprzęt do C64 lub Amigi, będący jakimś fragmentem historii, kupuję go i zanoszę do mojego „pokoju nostalgii”, który sam zbudowałem z kawałków, w nowym domu. Mam zamiar zachować w nim, przed zapomnieniem, wspaniałe sprzęty z lat 80-tych i 90-tych, takie jak komputery, magnetowidy, magnetofony i czasopisma lub książki komputerowe.

Moje inne zainteresowania dotyczą wszystkiego co jest związane z tworzeniem filmów, efektów specjalnych, tworzenia muzyki (tylko na prawdziwych instrumentach), technologii, tworzeniu czegoś pod różną postacią, a w szczególności na projektowaniu wnętrz. Moja strona internetowa lepiej przedstawia moje hobby, niż krótkie opisanie tego w kilku zdaniach. Dużo z moich osiągnięć jest także dostępnych pod postacią filmów na Youtube. Wystarczy tylko wspomnieć moją własnoręcznie zbudowaną dwumetrową wysokość **„Multi Arcade Machine”**. To naprawdę sprzęt nie z tej ziemi.

Po skończeniu szkoły średniej, rozpocząłem rok studiów na wydziale instalacji elektronicznych, który to kierunek był najbardziej zbliżony do moich zainteresowań z lat 90-tych. Jednak skoro nie jestem matematykiem, nie zaliczyłem końcowego egzaminu zawierającego właśnie matematykę i różne wzory matematyczne. Jednak nie było to dla mnie żadnym zaskoczeniem. Zawsze wydawało mi się, że jestem raczej praktykiem, który lubi pracować własnymi rękoma, a nie tylko umysłem.

Po tym roku dostałem pracę jako początkujący elektronik, pracujący przy produkcji i serwisie dla firmy transportowej i innych lokalnych firmach elektronicznych. Około 1996 roku pracowałem jako dostawca przesyłek przez trzy miesiące, a później zostałem zatrudniony jako magazynier w tej samej firmie, gdzie pracuję do dzisiaj.

Dziwnym trafem, pierwsza firma w której zacząłem pracować, została wchłonięta przez przedsiębiorstwo, w którym pracuję dzisiaj. Można powiedzieć, że w przeciągu tych wszystkich lat, pracowałem dla jednej firmy, począwszy od 17 roku życia. Osobiście nie jestem spontaniczny i nie staram się zrobić wszystkiego. Jestem raczej łatwowiernym psem. Sport i sporty ekstremalne nie są moim zainteresowaniem, tak samo jak szybkie samochody. Ogółem wszelkie niebezpieczne dziedziny życia, nie są w moim stylu.

Od 2000 roku zostałem zatrudniony jako grafik/projektant stron internetowych w firmie „Hatteland Display”, produkującej komputery nawigacyjne dla statków. Później do mojego zakresu działań, zostało dodane programista HTML, Basic, Dokumentator i grafik 3D. Do miejsca pracy mam jakieś dwie minuty drogi. Nie mam za bardzo czasu, żeby jeździć godzinę do pracy każdego dnia. Wolę ten czas przeznaczyć na bardziej kreatywne zajęcia w domu.

Czy mógłbyś opisać swój projekt „Multi Arcade Console”. Jakie były jej możliwości? Jak zdobyłeś lub może sam stworzyłeś oprogramowanie do niej. Czy ktoś Ci pomógł, czy też był to wyłącznie twój projekt.

Kiedy byłem dzieckiem (właściwie wciąż nim jestem), zawsze pragnąłem mieć swoją własną konsolę do gier w stylu automatu do gier w swoim domu. Granie i patrzenie na te maszyny w latach 80 i 90-tych zawsze wywoływało u mnie dreszczyk emocji. Grafika była poza możliwościami jakie oferował C64 i Amiga w tamtym czasie, ostatecznie tak to odbierałem. Nie wiem dlaczego ale te piksele i sposób w jaki tworzyły grafikę i animację zawsze wprawiało mnie w zdumienie. Skalowanie sprajtów w zawrotnym tempie było tym co w nich kochałem. LEGO też mnie przyciągało swoją „kwadratowością” :)

Minęło parę lat i pomysł budowy własnego automatu do gry powrócił podczas Bożonarodzeniowej przerwy w 2003 roku. Czy to było możliwe? Czy byłem w stanie temu podołać? Zacząłem poszukiwania opisu wykonania obudowy, i wtedy zdałem sobie sprawę że oprócz mnie istniało wiele innych osób, które tak jak ja, próbowały zbudować własne konsole. To była olbrzymia grupa, która traktowała to zajęcie jako hobby. Ale zaraz, dlaczego w sumie nie mogłem po prostu kupić starego dobrego automatu do gier zamiast męczyć się z własnym?

No cóż, sprzęty te to dzisiaj rzadkość, dodatkowo są drogie ze względu na stosowane w nich stare (niestosowane obecnie) rozwiązania elektroniczne, i dodatkowo posiadały tylko 1 grę. Tak więc coś musiałem stworzyć, coś swojego, coś co mogłoby przy użyciu kilku emulatorów zapewnić zróżnicowaną rozrywkę. Nie wspominając już o bardzo wyjątkowym własnym oprogramowaniu do uruchamiania gier, menu w stylu 3D Matrix z mnóstwem funkcji do sterowania, wyszukiwania i zaznaczania najlepszych gier. To wszystko było tylko fantazją..... i stało się rzeczywistością.

Przez 6 m-cy stworzyłem mnóstwo planów, łącznie z kompletnym modelem 3D obudowy maszyny z moich wizji nazwanej **SOMAC**. To nie miał być po prostu kolejny automat do gier zbudowany w domu. Moim celem było stworzenie maszyny, która byłaby tymi wszystkimi automatami razem. Chciałem aby miała mnóstwo przycisków, miejsce do wrzucania monet i obracany ekran TV pozwalający na komfortowe granie w gry, które wymagały poziomego ułożenia ekranu. Tak aby mogły powrócić w pełnej chwale jak za dawnych lat.

Tak... zawsze nie mogłem przestać myśleć o pewnych unikalnych gadżetach i wyjątkowych rozwiązaniach elektronicznych, które sprawiłyby że mój automat będzie jedyny w swoim rodzaju.

Mnóstwo złożonej elektroniki oraz rozwiązań urządzenia zostało przeze mnie wymyślonych i zapisanych. Dziś, kiedy cały projekt jest już ukończony mogę śmiało stwierdzić że 95% tych pomysłów zostało z sukcesem zaimplementowanych w moim projekcie. Składał się on z ogromnej liczby osobnych części, które musiały współpracować jak jedna duża całość. Chciałbym nadmienić że nie uczyłem się o zastosowaniu i użytkowaniu żadnej z nich. Nie jestem stolarzem ani elektronicznym geniuszem, ale to co potrafi znieść moje ograniczenia to stanie twarzą w twarz z problemem. Bezsenne noce i ciągłe myślenie o SOMACu było całym moim życiem przez 6 następnych miesięcy. Ale projekt nie stanął w miejscu. Dopiero po 2 latach konstruowania zorientowałem się jak

szalony to był pomysł i co tak naprawdę udało mi się stworzyć.

To była ciągła walka z problemami zarówno przy projektowaniu jak i konstruowaniu urządzenia. Miało miejsce mnóstwo poprawek i opóźnień. Jednak jestem cierpliwy, nie uznaję porażek tylko zwycięstwo i osiągnięcie celów.

Prawie 2 lata później udało mi się zrealizować swój projekt budowy najbardziej rozbudowanej konsoli skonstruowanej w warunkach domowych. Nazwałem ją **Stone Oakvalley's Multi Arcade Console 2000**.

Dlaczego 2000? Po to żeby zachować ducha tamtych czasów. W latach 80 i 90-tych, wiele urządzeń i gier (które w założeniach miały być wspaniałe) miało dodane 2000. Była to ludzka wizja technologii, która miała się ziścić do roku 2000. Niektórzy nawet dodawali w nazwie 3000.

Dlaczego SOMAC ma w nazwie multi? To proste. Składa się on z programowego emulatora zainstalowanego wewnątrz, który emuluje wiele klasycznego sprzętu jak np.: Classic Arcade Machines, Commodore 64, Commodore Amiga, Nintendo, Super Nintendo, Nintendo 64, SEGA Genesis, SEGA Master System, SEGA Game Gear, Gameboy Advance, NEC TurboGrafx 16, Atari 5200/7800/Lynx & SONY Playstation.

30% automatu zostało zrobione z porzuconych lub częściowo działających części złożonych w całkiem nowy sposób. Oprogramowanie sterujące zostało stworzone przeze mnie i nazwane „**3D Matrix Core Center**” i „**Touch Center**”. Oprogramowanie „**3D Matrix Core Center**” składa się z posortowanych w kolejności alfabetycznej gram. Posiada możliwość wyszukiwania, dodawania do ulubionych, zmiany gatunku lub usunięcia zaznaczonej gry w razie potrzeby. Wszystko to jest przedstawione na trójwymiarowej matrycy, która prezentuje się wspaniale. Operować można za pomocą klawiatury, trackbala lub joysticka. Wszystko pracuje na głównej maszynie która kontroluje każdy z pozostałych emulatorów takich komputerów jak: Arcade Machines Emulator: MA-ME(tm), Commodore 64 Emulator: WinVice, Commodore Amiga Emulator: WinUAE, Nintendo Emulator: VirtuaNES, Super Nintendo Emulator: SNES9x, Nintendo64 Emulator: Project64, Sega Genesis/Game Gear/Sega System Emulator: Kega Fusion, Gameboy Advance: VisualBoyAdvance, NEC Turbografx-16 Emulator: MagicEngine, Atari 5200/7800 Emulator: MESS, Atari Lynx Emulator: Handy and Sony Playstation 1 Emulator: ePSXe. „**Touch Center**” zapewnia łatwy dostęp do wszystkich emulatorów sprzętowych. Uruchamiany jest na

oddzielnym ekranie dotykowym i komunikuje się z PC za pomocą portu RS-232.

Zarówno sprzęt jak i oprogramowanie zostało wykonane w 100% przeze mnie. Całość działa pod kontrolą WindowsXP. Wymagało to takich rzemiosł jak: stolarza, grafika i grafika 3D, inwencjonalności, programisty oraz wielu innych prac porównywalnych do jakiejś firmy, zajmujących się podobnymi rzeczami, ale wykonanych przez jedną osobę.

Konsola zawiera 26188 gier. Każda ze zrzutem ekranu i odpowiednio skatalogowana. Całość podzielona na 16 różnych emulatorów. Baza danych gier została nazwana SGIL (SOMAC Game Intelligent Listing) i została stworzona po rozczarowaniu, jakie przeżyłem przeglądając różne strony o grach. Brak tam było informacji, które mógłbym wykorzystać w mojej bazie.

Podsumowując, projekt SOMAC składa się z:

- najbardziej spektakularnego automatu do gier na tej planecie;
- niesamowitego wyglądu zewnętrznego automatu do gier;
- wyglądu zbudowanego od podstaw;
- własnego oprogramowania - „3D Matrix Core Menu System”;
- stworzonej specjalnie dla konsoli bazy danych nazwanej „SGIL - SOMAC Game Intelligent Listing”;
- najbardziej kompletnej bazy i najbardziej użytecznej, spośród wszystkich baz o grach na całym świecie;
- logicznej bazy danych składającej się z tytułu, zrzutu ekranu, gatunku;
- zawartość bazy stanowi 26188 gier obejmujących 16 klasycznych maszyn;
- 4 zestawy kontrolerów, umieszczonych na obrotowym blacie;
- kontroli bazy danych gier za pomocą ekranu dotykowego;
- całkowicie zautomatyzowana kontrola programów emulujących poszczególne komputery.

Osoby chcące bliżej zapoznać się z projektem powinny zajrzeć na strony:

www.stone-oakvalley-studios.com/index-so-mac_id.php
www.youtube.com/watch?v=K6bUeQ3Zpco

Wspomniałeś że interesowałeś się kinem. Czy kiedykolwiek reżyserowałeś - montowałeś jakiś film. Jeśli tak to czy mógłbyś opisać na jakim sprzęcie, oprogramowaniu pracowałeś i co cię inspirowało.

Nigdy nie reżyserowałem ani nie zajmowałem się postprodukcją jakiegokolwiek komercyjnego filmu. Nigdy też nie uczestniczyłem nawet w średnio udanych produkcjach.

Mój pierwszy kontakt z filmami był taki jak w przypadku większości amatorów. Najpierw pomysł, a potem próba jego realizacji w postaci krótszych lub dłuższych filmów kręconych w warunkach domowych wraz z przyjaciółmi. Wszystko to przy ograniczeniach czasowych oraz wszystkich innych zasobów, ale za to z wielkim poświęceniem oraz wyobraźnią. Filmy które nakręciłem zatytułowałem „**Knut Mollvik, A tragedy/A interview with an artist**”, „**Violence amongst neighbours**”. Był także dłuższy zwiastun filmu który mieliśmy w planach nakręcić pt. „**Valkyrie (Vampyris)**”, oraz pomysł i koncept na „**Ylva, The girl, The woman, the wolf**” oraz „**Moon Fever**”.

„**Knut Mollvik**” był kręcony na sprzęcie VHS w 1997. Potem w roku 2000 został ponownie zmontowany i poprawiony w programie Adobe Premiere 4. Do zrzucenia obrazu posłużył grabber DV500. „**Violence amongst Neighbours**” był kręcony w technologii VHS, VHS-C i S-VHS. Następnie w całości zmontowany w „Adobe Premiere 6.5” z pomocą LightWave’a 3D, Commotion i przy użyciu programu Icarus Motion Tracking z DV500 oraz dodatkowym monitorem. „**Valkyrie**” z kolei był kręcony na sprzęcie DV i został zmontowany w Adobe Premiere 6.5/7 pozostałe wyposażenie bez zmian.

Moje związki z prawdziwym kinem opierały się na krótkim materiale filmowym do filmu „Wide Blue Yonder” mającego swoją premierę w 2010 roku. Spotkałem się wtedy z takimi gwiazdami kina jak Brian Cox, Wally, Lauren Bacall, James Fox, Hege Schøyen (z którym zjadłem kolację w towarzystwie dystrybutora Bjorga Velande oraz reżysera Roberta Younga). Z pozostałych sław można wymienić Ingrid Bolso Berdal, Kare Conradi, Sverre Anker Ousdal i Arne Oddvar Husby.

Dla kina pracowałem w wolnym czasie. Pomagałem w pracach nad broszurami, robiłem kompilacje dvd, projektowałem wstępne szaty graficzne plakatów, wysyłałem newslettery e-mailem, pomagałem przy projektowaniu stron do filmów, wszystko to wspólnie z firmami dystrybucyjnymi o zasięgu międzynarodowym takimi jak BV Films, Euromax and Parkland Pictures. Wszystko trwało od 2004 do 2007 roku. (w 2007 dołączyli mój pomysł na film „Valkyrie (Vampyris)” do swojego harmonogramu i posłali mój plakat na Festival Filmów w Can. Wszyscy mogli go obejrzeć).

Projekt „Valkyrie (Vampyris)” został zawieszony w momencie gdy mój przyjaciel i jednocześnie scenarzysta Tom Birger Jenssen zmarł w 2006 r. na atak serca. Wraz z jego córką kontynuowaliśmy go, ale nigdy nie udało nam się stworzyć jakiejś zadowolającej wersji

finalnej. Był on coraz bardziej wypierany przez inny wariant całej historii nazwany „Ylva”. Z powodu braku czasu ten projekt został również odłożony.

Różne odnośniki:
www.stone-oakvalley-studios.com/shortfilm_index_van.php
www.stone-oakvalley-studios.com/index_val-kyrie.php
www.ylvathewolf.com
www.stone-oakvalley-studios.com/index_films_introduction.php

YouTube:
www.youtube.com/watch?v=G0de0shDagI
www.youtube.com/watch?v=o1DbtBJOueI
www.youtube.com/watch?v=6PCmTnzfGVk
www.youtube.com/watch?v=uisGrQUbkVQ
www.youtube.com/watch?v=SuYUTLnTLss
www.youtube.com/watch?v=tWmahGYqkBQ
www.youtube.com/watch?v=Y77J_0yDkCQ
www.youtube.com/watch?v=OyW6mXaYsFM
www.youtube.com/watch?v=AH7_GbNiKBg

Tworzysz ogromne archiwum nazwane SOASC=. Czy możesz opisać, jak doszło do tego projektu i czym on właściwie jest?

Idea na stworzenie wersji MP3 archiwum HVSC naszła mnie, kiedy natknąłem się na stronę, na której było porównanie nagranej emulowanej wersji utworu na SID’a (Sid-play2w) i nagranej z oryginalnego SID’a z komodorka. Utwór nazywał się „Gloria”, a autorem jest Mitch i Dane. Przez wiele lat cieszyłem się ze słuchania emulowanego SID’a i myślałem, że tak właśnie brzmiały te dźwięki z mojego dzieciństwa... Myliłem się i to bardzo.

Znając przez lata HVSC, przeszukałem internet w kierunku „mp3-jkowej wersji archiwum HVSC”, jednak nie mogłem znaleźć żadnego większego archiwum. To na co się natknąłem, było małymi bazami z kilkoma utworami zamieszczonymi w internecie z różnych powodów. Wspomniałem mojemu przyjacielowi Waxheadowi, że byłoby wspaniale nagrać całe archiwum HVSC w formie MP3 i pokazać

mu wielką różnicę pomiędzy emulacją, a prawdziwym SID’em przez komunikator MSN. Wybuchł jak bomba jądrowa, mówiąc „**TAK, ZRÓB TO, ZRÓB TO, ZRÓB TO STONE!**”. Po tym wiedziałem, że takie przedsięwzięcie znajdzie wielką aprobatę i po pięciu minutach od ściągnięcia HVSC#45 zabrałem się do pracy.

Po tym wygrzebałem swojego starego C64 i zrobiłem test porównując różnicę między emulacją a oryginałem. W rzeczywistości jest ogromna. Testowany był chip w wersji MOS6581R4. Zacząłem szukać możliwości automatyzacji procesu nagrywania, bez ingerencji człowieka... Nie obeszło się bez kilku eksperymentów i wielu poszukiwań, dopóki nie znalazłem rozwiązania. Nie obeszło się bez kilku dodatkowych elementów, trochę własnej elektroniki, kabli i własnoręcznie napisanego oprogramowania.

Kolejną inspiracją do wykonania tego projektu był fakt, że HVSC jest bardzo dobrze zorganizowaną bazą danych, bez różnych dziwnych błędów lub komplikacji. Tak więc kłaniam się w kierunku grupy HVSC za stworzenie tak wspaniałej bazy danych. To była porządna podstawa do dalszej pracy. Stworzyłem także własną bazę danych z narzędziami do analizowania i przeglądania długości wszystkich SID’ów, w celu stworzenia własnej wersji plików INI generowanych podczas procesu nagrywania.

Dzisiaj archiwum zawiera wersje PAL i NTSC dla układów MOS6581R2, MOS6581R4 i CSG8580R5. Ukazuje to różnice pomiędzy tymi wersjami układów. Wersja R4 ma porządne filtry, R2 ma otwarte filtry, a R5 po prostu jest jaki jest. Nie znam żadnych różnic dla tej wersji układu w stosunku do pozostałych. Pozostała jeszcze wersja R3, jednak podczas realizowania tego projektu, natrafiłem na kilka problemów i sytuacji, które spowolniły proces jego realizacji. Musiałem np. ponownie nagrywać uszkodzone utwory, nie wspominając o nagrywaniu tych samych wersji w NTSC, oraz ciągle uaktualniać swoją bazę względem



HVSC. Doprowadziło to do wydłużenia całego przedsięwzięcia do 2,5 roku, dopóki wszystkie znane błędy nie zostały usunięte. Projekt zakończyłem w kwietniu 2009 roku.

Sprzęt do nagrywania występował w dwóch wersjach. Pierwszą był trzymetrowej długości stół, wypełniony elektroniką, który działał przez jakieś 8 miesięcy. Drugą wersją była kabina nazwana „Commodroid”, która zajmowała mniej miejsca, jednak jej bolączką była ciasnota przy zmienianiu konfiguracji czyli różnych układów SID. W końcowej opcji, przewody, zasilacz PC i dysk twardy nie zdały egzaminu. Tak samo było z połączeniem między PC a C64. Całe okablowanie osiągało 50% swojej wydajności dla HVSC#49. Commodoid był już zmęczony, tak samo jak i ja.

Muzyka była nagrywana w formacie WAV, a później kompresowana do formatu MP3 w wysokiej jakości. Miejsce zajmowane przez archiwum WAV/FLAC okazało się za duże w 2006, a ja nie miałem ochoty tworzyć studia Dolby-Digital-Michael-Jackson, żeby mieć muzykę wysokiej jakości. Chciałem stworzyć to wszystko jak najmniejszym kosztem. Dodatkowo pragnąłem stworzyć archiwum zawierające muzykę zbliżoną parametrami do tego, co słyszeliśmy siedząc jako dzieciaki przed marnej jakości telewizorem i mając podłączony mierny sprzęt audio. Tak właśnie dla mnie działa nostalgia, właśnie do powrotu do tych wszystkich zgrzytów i szumów, jakie wtedy się słyszało, a nie do czystych brzmień wydobywanych z drogiego sprzętu.

Na początku całe archiwum było przeznaczone do mojego własnego użytku. Hostowanie 450GB mogło okazać się problematyczne, zważywszy na fakt, że chciałem to robić bez płacenia komukolwiek. Na szczęście znajomy ma połączenie 10MBps i zdecydował się na udostępnienie tego łącza światu. Niestety to nadeszło po próbach korzystania z płatnych serwerów, które okazały się wielkim błędem. Waxhead napisał bazę danych w C/PHP, plując na czysty html z różnych powodów, takich

jak kontrola nad materiałem, oferowanie szybszego wyszukiwania, żadnej bzdurnej licencji itp. Archiwum zawiera dzisiaj 154473 plików MP3 upchanych na przestrzeni 450GB, dodatkowo znajduje się kopia archiwum SID o nazwie HVSC#49. Nagrywanie rozpocząłem od HVSC#45. Planowałem zakończyć je po numerze 50, jednak wspomniane problemy ze sprzętem, przeprowadzka do nowego domu wymagającego remontu i kończenie projektu SO-AMC=, okazało się czasem dla definitywnego zakończenia SOASC= w kwietniu 2009 roku.

Na razie nie planuję wznowienia nagrywania w celu poprawienia pozostałych błędów. W tym celu musiałbym robić większość rzeczy ręcznie, lub odtworzyć cały sprzęt na nowo. Jeśli kiedyś zajmę się tym ponownie, sprzęt będzie zawierał kilka C64 pracujących w tym samym czasie, w których będą różne wersje SIDów, szybsze i łatwiejsze wczytywanie plików i dołączone zgrywanie muzyki do formatu FLAC.

Kolejnym uczuciem jakim darzę ten projekt, jest zachowanie wspomnień dla znanej muzyki z lat dziecięcych. Muzyka tworzona w dzisiejszych czasach na C64 nadal jest wspaniała, jednak nie ma tej nostalgii, ponieważ została stworzona niedawno. Być może za 10-20 lat, muzyka tworzona dzisiaj będzie mile wspomniana, jednak nie jestem pewien czy to akurat ja będę się zajmował jej zgrywaniem. Dobry projekt, który „umarł” może zostać wznowiony przez kogoś, kto będzie chciał dodać co nieco od siebie. Być może są już ludzie, którzy w tajemnicy nagrywają muzykę z 10-ciu różnych wersji SIDów do formatu FLAC, w celu powiększenia archiwów SOASC= lub SOAMC=. Osobiście jestem szczęśliwy ze stworzenia SO-ASC= i pozostawiam to jako dobry przykład dla dedykacji i respektu do komputerów, które są mi i innym ludziom bardzo bliskie.

Dziękuję firmie Commodore za stworzenie najlepszego komputera domowego wszechczasów!! Chciałbym również podziękować całej grupie HVSC za prowadzenie niezwykłego projektu i na koniec chciałbym wy-

razić swój szacunek dla całej grupy SOASC=.

Jak wspominasz tamte czasy kiedy komponowałeś na Amidze i brałeś czynny udział w demo scenie? Pamiętasz może nazwy grup do których należałeś? Gdzie możemy znaleźć twoje produkcje?

Dema przy których realizacji brałem udział, i które zostały ukończone to „PCPuke”, „Powerdeath” and „The Wares From The Gathering 1994 intro”. Jedyną grupą do której należałem w latach 1993-1994 to „Principes”. Pozostałe grupy składały się zasadniczo z moich przyjaciół lub tylko ze mnie.

Na tych stronach próbowałem zgromadzić swoją historię amigowską:

www.stone-oakvalley-studios.com/wordpress/?page_id=457

www.stone-oakvalley-studios.com/wordpress/?page_id=905

www.stone-oakvalley-studios.com/wordpress/?page_id=195

www.stone-oakvalley-studios.com/wordpress/?page_id=526

Odnosiniki do plików:

www.pouet.net/prod.php?which=14500

www.pouet.net/prod.php?which=14499

www.bitworld.bitfellas.org/demo.php?id=5190

www.fonix.dyndns.org:40000/soamc/index.php?sb=SOAMC&ss=&sf=&sy=&sc=demonoid+productions&srd=&sd=&sr=&sx=&sl=

Co to jest Commodroid? Czy mógłbyś przybliżyć nam ten projekt?

Commodroid był nazwą drugiej wersji systemu nagrywającego dla projektu SO-ASC=. Zawierał całą elektronikę z wersji 1-szej, umieszczoną w szafie, która zajmowała mniej miejsca, niż pierwsza wersja rozłożona na 3-metrowej długości blacie. Zawierał on: 1x mydelniczka z układem SID 6581, 1x C64C z układem 8580, dwóch PC (33 i 233 MHz) pracujących pod DOSem z programem 64HDD, który stanowił serwer plików *.prg dla C64, dwóch pecetów (800 i 933 MHz) pracujących pod WinME, które służyły do nagrywania muzyki. Uruchomione na nich było autorskie oprogramowanie nagrywające (SIDREC). Dodatkowo zamontowane były dwa 12-to calowe ekrany dotykowe, jeden 7-mio calowy monitor, dwa zmontowane własnoręcznie interfejsy klawiatury do „C64 PAR Relay” - pracujących na zmianę, dwóch zestawów XE1541, dwóch przewodów Audio/Video do C64 i 12-tu gniazd zasilających 220V.

Wygląd urządzenia przypominał projekty robotów z lat 60-70'tych. Stąd właśnie wzięła się nazwa Commodroid. Właściwie to nazwę wymyślił członek naszego forum gdy zobaczył fotografię tego „stworu”.



Odnosiłki do plików:
www.6581-8580.com/soasc_how.php
www.youtube.com/watch?v=H7V2piLBjj8
www.youtube.com/watch?v=vz2AgjH0adY
www.youtube.com/watch?v=YteeUoLY6s0
www.youtube.com/watch?v=qTEDNWmCz3U
www.youtube.com/watch?v=k_OO6qWpjb0
www.youtube.com/watch?v=TkrKbKt4qyI

Opowiedz coś więcej o projektach SOASC= i SOAMC=. Co cię zainspirowało, w jaki sposób działają, może pamiętasz jakieś ciekawe historie, problemy bądź trudności związane z tymi projektami?

Projekt polegał na nagrywaniu utworów z oryginalnego C64. Ważne było aby wszystko odbywało się na oryginalnym sprzęcie, bez jakichkolwiek prób poprawy dźwięku w jakikolwiek sposób. Wyjątek stanowiła jedynie redukcja pewnego subtelного szumu generowanego przez wejście audio SID'a do GND. Taki sposób połączenia powodował pewien szum na łączu z wyjściem audio SID'a przez port Audio/Video z tyłu C64, które to było używane w tym projekcie. Wszystkie pozostałe elementy na C64 były pozostawione dokładnie w takim stanie jak wyszły z montowni. Powstały liczne rozszerzenia dla SIDA takie jak stereo sid, miksowanie dźwięków z układów 6581 i 8580, jednak ja nie chciałem wykonać żadnego z nich. Dźwięk musiał być prawdziwy i autentyczny, dokładnie tak jak w latach świetności C64. Projekt obejmuje archiwum 154473 plików w formacie MP3 zgranych z układów 6581R2, 6581R4 i 8580R5. Ponieważ mieszkam w Norwegii początkowo dźwięk odgrywany był przez commodorki w systemie PAL. Potem na ebay'u kupiłem C64 w wersji NTSC i utwory odtwarzane na nim też weszły w skład archiwum. Format MP3 wybrałem ze względu na najwyższy stopień kompatybilności między różnymi platformami. Format ten jest także najlepiej supportowanym formatem na dzień dzisiejszy.

Miałem do wyboru dwa cele dla tego projektu. Albo oddanie poczucia dźwięku, jaki brzmiałby przy podłączeniu C64 do współczesnego sprzętu audio, lub do nostalgii związanej z przypomnieniem sobie przesiadywania przed zwykłym telewizorem i słuchania dźwięku wydobywającego się z głośnika bez basów i wysokich tonów, oraz masą innych ograniczeń które były udziałem sprzętu z lat 80-tych.

Projekt ten od początku był bardzo zorientowany na zachowanie autentycznego brzmienia. W ten sposób miał przypominać dawne czasy, w których rodziło się mnóstwo wspaniałych talentów tylko dlatego że ludzi wproważyły w zdumienie możliwości muzyczne i graficzne C64. Taka jest mniej więcej również moja historia. W tamtych dniach gdy używałem Commodore, in-

teresowałem się grafiką, muzyką i programowaniem, które to zainteresowania stały się obecnie moją pracą. Zajmuję się grafiką komputerową, projektuję strony www, wykorzystuję również grafikę 3D oraz zajmuję się programowaniem. Wszystko to za sprawą C64, dodatkowo zadania związane z moją pracą, są moim hobby. W ten właśnie sposób chciałem również uhonorować firmę Commodore za stworzenie tej wspaniałej maszyny. Jej komputery rozwinęły moją wyobraźnię na wielu różnych poziomach.

Właściwie, tak jak mnóstwo innych ludzi, byłem szczęśliwy mogąc słuchać utworów z C64 na emulowanym SIDie na Amidzie i PC. To było w latach 1987-99. Kilka lat później (2002-3) zaczęło mi brakować brzmienia SIDA i tak jak inni ludzie zacząłem słuchać emulowanych dźwięków na programie SIDPLAY2, byłem z tego powodu bardzo szczęśliwy. Tak było do roku 2006 gdy trafiłem na stronę z 3 czy 4 utworami, które prezentowały poważne różnice w dźwiękach oraz filtrowaniu dźwięku pomiędzy emulatorami, a rzeczywistym C64. Zachęciło mnie to do poszukiwania bardziej wiarygodnych utworów i w końcu wstukałem w przeglądarkę Google frazę „HVSC as MP3”. Zaskoczyło mnie to, że Google zwrócił pusty zbiór jako wynik wyszukiwania. Moja potrzeba słuchania utworów z C64 w oryginalnym brzmieniu z lat 80-tych nie została zaspokojona.

Zasadniczo chciałem znaleźć kogoś, kto już wcześniej zrealizował taki projekt, ale nie mogłem nikogo takiego znaleźć. Z tego względu byłem naprawdę szczęśliwy że udało mi się wymyślić coś czego nikt wcześniej nie zrealizował, coś co pierwotnie miałem zamiar zrobić tylko dla siebie samego. Zdałem sobie sprawę, że projekt ten może być upragnioną rzeczą o jakiej marzyli pozostali fani C64. Cały czas pamiętając, że projekt dotyczył najlepiej sprzedającego się komputera domowego, byłem przekonany co do słuszności umieszczenia go w necie, gdzie każdy chętny mógłby sobie go pobrać. Nawet jeżeli przestrzeń zajmowana na dysku osiągnie wielkie rozmiary, byłem przekonany że wielu ludzi którzy dotychczas zadowalali się emulowanym SIDem powita z radością mój projekt. Chciałem również uświadomić ludzi co do jakości odtwarzanych utworów, tak jak ja zostałem uświadomiony.

Projekt SOASC= był dla mnie olbrzymim technologicznym wyzwaniem i zarazem potrzebą przekonania się czy jestem w stanie podolać wyzwaniom z nim związanym. Wyciągnąłem swojego komodorka i przeprowadzałem testy polegające na odsłuchiowaniu brzmienia oraz pierwsze niezdarne próby nagrań, aby przekonać się czy naprawdę występują opisane różnice w brzmieniu. Nigdy nie spodziewałem się że SID, a szczególnie 6581,

odtwarzał tak nieczysto dźwięki dla tak wielu różnych wariacji filtrów. Emulowany 8580 jest prawie tak dobry jak prawdziwy układ, ale jak powszechnie wiadomo 6581 miał w pewnych obszarach problemy z generowaniem czystego dźwięku, a także miał kilka poważnych wad konstrukcyjnych. Ale na jego problemy można spojrzeć także z innej perspektywy, jako na unikalną charakterystykę dźwięku który mógł być wygenerowany tylko przez ten układ i żaden inny. Czasami rzeczy nie muszą być doskonałe aby za takie zacząć je uważać, ponieważ każdy ma swoją skalę doskonałości.

Kolejnym technicznym wyzwaniem było takie skonstruowanie całego zestawu do nagrywania, aby powstała w pełni automatyczna maszyna działająca w pętli, całkowicie bez lub też z niezbędnym minimalnym udziałem człowieka polegającym na klikaniu, pisaniu lub nadzorowaniu działania całości. Obróbka każdego utworu przy takiej ich liczbie (około 115000) nie byłaby możliwa przez człowieka. Stworzyłem więc swoje własne oprogramowanie w PureBasic, łączące porty PAR z PC na karcie przełącznikowej stworzonej przez Sveina Engelsgerda. Do tego dołączyłem pewne klawisze z klawiatury (na kablu klawiatury) i pin resetu z user portu C64 w celu uzyskania „zdalnego” sterowania komodorkiem. Wszystkie utwory zrippowane w archiwum HVSC były konwertowane do uruchamialnych plików prg za pomocą PSID64 autorstwa Rolanda Hermansa. Utwory ładowano przy użyciu HDD64 stworzonego przez Nicholasa Coplina który połączony był przez port PAR drugiego PC, działającego pod kontrolą DOS'u, przez port Serial C64 przy użyciu kabla XE-1541. Utwór był z kolei nagrywany przy użyciu innego peceta. Dla każdego C64 oddzielnie postawiony był serwer plików i serwer nagrywający. Ponieważ nagrywałem 2 wersje utworu z SIDA 6581 i 8580 wymagało to w sumie 4 pecetów i 2 C64 upakowane razem na 3 metrowej długości tablicy. Była tam upakowana sama goła elektronika. Całość potem została przeprojektowana aby weszła do szafki. Wtedy też powstała nazwa całego urządzenia Commodroid. Nagrywał on także więcej wariantów brzmień układu 6581, łącznie z utworami pisanymi pod maszyny działające w trybie NTSC. Nagranie zakończyło się na 49 wydaniu bazy HVSC.

C64 był dla mnie atrakcyjny ze względu na połączenie muzyki z grafiką, ale nigdy ze względu na samą muzykę. Miałem obsesję na punkcie elementów graficznych poruszających się na ekranie w sposób zsynchronizowany z muzyką. Zadziwiającym było jak ludzie w swoich demach starali się robić coś w stylu rave/techno z początków MTV prawie dekadę wcześniej, niż styl ten stał się powszechnie znany za sprawą MTV. Kolejną rzeczą która wywarła na mnie

olbrzymie wrażenie był to moment kiedy usłyszałem głosy w grze „Ghostbuster” oraz samplewaną perkusję w demie Lukullusa pod tytułem „It's Drum Time”. Te dwa momenty wywarły na mnie olbrzymi wpływ. Nigdy bym się nie spodziewał, że te dźwięki są generowane przez komputer. Kolejną rzeczą, która bardzo mi się podobała była gra pt. „Last Ninja” i sposób w jaki muzyka budowała specyficzny nastrój w zależności od aktualnej sytuacji gracza. Była to kombinacja odkrywania tajemnic, wspaniałej grafiki i muzyki mającej ogromny wpływ na klimat całej rozgrywki. Wszystko to wywołało u mnie potrzebę „odkrycia” kolejnego etapu, aby doświadczyć tych uczuć ponownie, ale już z nowymi tajemnicami, poznać grafikę następnego etapu i aby posłuchać jeszcze doskonalszej muzyki, to mnie ciągle kierowało do osiągnięcia celu. Czułem się wtedy jak prawdziwy bohater.

Dziwiłem się także jak ludzie, tacy sami jak ja, potrafili tworzyć wspaniałe grafiki i muzykę. Patrzyłem na koderów dem jak na nadludzi i miałem nadzieję że pewnego dnia będę taki sam jak oni. Wydawało mi się, że firmy produkujące gry, nie zatrudniały zwykłych ludzi tylko cyborgi które tworzyły tę fantastyczną oprawę muzyczną i graficzną. Demka i gry były oczywiście tworzone przez zwykłych ludzi, ale fajnie było wtedy jako dziecko tak o tym myśleć. Dzieci mają swój własny sposób na wytłumaczenie sobie, jak świat jest zorganizowany. Kiedyś nawet myślałem że Rob Hubbard napisał demo, w którym użyto tylko jego utworu. Ale wraz z dorastaniem coraz bardziej zdawałem sobie sprawę jak to się wszystko kręci.

Wierzę że ta muzyka wciąż rozbrzmiewa u ludzi ponieważ mają oni te same odczucia, które ja miałem. Wrażenia, fascynacje i wspaniałe chwile w życiu dziecka/nastolatka i fakt że czasami trzeba było wyobrażać sobie pewne elementy gier i środowiska w jakich się toczyły. Dzisiaj wszystko jest możliwe do zwizualizowania, tym samym zostajemy pozbawieni możliwości odgadnięcia pewnych elementów świata gry. Muzyka generowana przez SID nie pozwalała nigdy na „wyobrażanie” sobie brakujących dźwięków. Patrząc z dzisiejszego punktu widzenia układ wciąż generuje unikalne i interesujące dźwięki dające zaskakująco dobre efekty w połączeniu z głosem ludzkim oraz innymi prawdziwymi instrumentami dla muzyków poszukujących innych brzmień, których sposób generowania dla większości pozostanie tajemnicą.

Czasami ludzie poszukują sposobów aby powrócić choć na chwilę do czasu kiedy byli dziećmi. Kiedy jedynym ich problemem była mama każąca kłaść się już do łóżka, błędy przy wczytywaniu programów z kaset i frustracja spowodowana nie ukończeniem danego etapu gry. Te długie weekendy z przyjaciółmi i grami na

C64 w swoim ulubionym fotelu i ten komfort... stanowią coś co ludzie chcą wspominać. Jest to niemożliwe dla tych którzy sprzedali wcześniej swoje komodoriki, a nie mają ochoty na pracę z popularnymi niegdyś formatami muzyki.

Dzięki projektowi SOASC=, aby posłuchać muzyki z prawdziwego C64 nie trzeba mieć żadnego dodatkowego sprzętu. Grafika z czasów C64 została dzisiaj znacząco poprawiona. Gdy przygodę z grami komputerowymi zacząłem niedawno to nie masz już możliwości ani miejsca na wyobrażanie sobie pewnych rzeczy w danych lokacjach, są one podane na tacy ze wszystkimi szczegółami i w jakości HD. Ale muzyka z SID'a zawsze była unikalna i nigdy nie zostanie zastąpiona przez jakiekolwiek dźwięki generowane przez układ muzyczny nowej generacji – jest po prostu jedyna w swoim rodzaju. Mamy emulator, które pozwalają w komfortowy sposób używać C64 na peciecie lub Mac'u, ale tak naprawdę nic nie zastąpi i nic nie można porównać z prawdziwym commodore, z dotykiem jego klawiatury, otwieraniem kieszeni magnetofonu i długim oczekiwaniem przyrywaniem zajądaniem różnych przysmaków i wsłuchiowaniem się w muzykę Ocean's Loadera.

Nie zdajemy sobie sprawy, że np. marne dźwięki w „Arkanoidzie” były dźwiękami prawdziwej perkusji puszczanej wstecz lub że Rob Hubbard zamierzał swoje utwory pt. „One Man and his Droid” przearanżować na utwór grany przez orkiestrę symfoniczną. Myślę, że kochamy słuchać C64, ponieważ SID generuje dźwięki niemożliwe do odtworzenia przez inne instrumenty.

Wydaje się, że w naturze ludzkiej leży przetwarzanie, tworzenie nowych wersji starych hitów tak, aby poczuć ponownie wspaniałe momenty towarzyszące wcześniejszym utworom muzycznym i filmowym. Pozwalają przeżyć obecnemu pokoleniu to, co sami kiedyś przeżywali, tylko teraz w unowocześnionej wersji. Wierzę, że C64 nie bazował na niczym wcześniejszym, była to świeża, niesamowita idea nie bazująca i nie poprawiająca niczego co było przed nim. Dlatego komputer ten wywarł tak ogromny wpływ na historię komputerów, ponieważ wyniósł standardy na kolejny wyższy, inny wręcz poziom. Dla SID'a kolejny poziom nigdy nie nastąpił. Nigdy nie wypuszczono jego godnego następcy. Tak myślę, że jeżeli coś zostanie zablokowane w czasie i nigdy potem nie będzie możliwe usprawnienie tego i wypuszczenie następnej bardziej zaawansowanej wersji, to takie coś będzie trwać na zawsze i nigdy nie zaniknie. To jest właśnie jeden z powodów, dlaczego ludzie ciągle kochają słuchać utwory z C64 na SID'zie, ten dźwięk jest po prostu unikalny. Czasami zastanawiam się, dlaczego muzyka z SID'a wprawia mnie zawsze w dobry nastrój,

podobnie jak w dzieciństwie. To jest tak jak z muzyką w cyrku, zawsze powoduje uśmiech na twarzy. Szczegółowe informacje o projekcie można znaleźć na stronie www.6581-8580.com, na filmach z YouTube i w sekcji FAQ.

Czy mógłbyś przybliżyć czytelnikom postać swojego przyjaciela Waxheada, który to miał pewien wkład w Twój projekt. Jak i kiedy się poznaliście?

Svein Engelsgerd, pseudonim Waxhead, wniósł dużo pomysłów do obu projektów SOASC= i SOAMC=. Wykonał również wiele niesamowitych nagrań, testował i odsłuchiwał nagrania i zaprojektował dla mnie urządzenie PAR Relay. W trakcie trwania projektu był odpowiedzialny za stworzenie forum obu projektów. Stworzył także dla nich wspaniałą, napisaną w językach C i PHP silnik bazy danych działającej online. Jej główną cechą było bardzo szybkie skanowanie i wyszukiwanie tekstu. Wszystkie zapytania zwracały wyniki po 0.39 sekundy.

Sveina poznałem podczas rozmowy telefonicznej w roku 1991 lub 92. Znalazł się w posiadaniu kopii jednej z produkcji naszej grupy pod tytułem „Redline Tools #1”. Spodobała mu się także jedna z moich pierwszych kompozycji na Amidze pt. „Piano Devil”, która zrobiona była na Soundtrackerze. Ciekawił go sposób w jaki ta kompozycja powstała. Poza tym zadawał mnóstwo różnych pytań dotyczących „Redline Tools”. Zaproponował także chęć współpracy w przyszłości. Od tego czasu miałem z nim stały kontakt telefoniczny, czasami spotykaliśmy się lub do kontaktu wykorzystywaliśmy modem w systemie ABBS lub MSN, a po 2000 roku tylko Internet. Svein mieszka około 25 minut jazdy samochodem (w zależności kto prowadzi) ode mnie i pracuje w tej samej firmie co ja. Dzięki temu mogliśmy utrzymywać ze sobą stały kontakt i wspominać stare czasy prawie co tydzień.

Odnosiłki do plików:
www.dirtcellar.net
fonix.dyndns.org:40000/soamc/index.php?sb=SOAMC&ss=piano+devil&sf=&sy=&sc=&sr=&sd=&sr=&sx=&sl=
www.stone-oakvalley-studios.com/wordpress/?page_id=457

Dziękuję za wywiad. Mam nadzieję, że jeszcze usłyszymy o kolejnych wspaniałych projektach zrealizowanych przez Ciebie, oraz że będą one dotyczyły komputerów stworzonych przez firmę Commodore. Dodatkowo życzę powodzenia w życiu prywatnym i zawodowym, oraz inwencji przy tworzeniu czegoś nowego, co jeszcze nie zostało wymyślone.

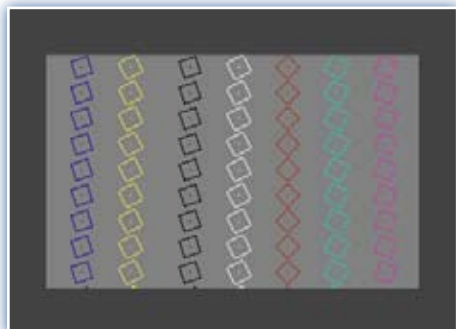
Wywiad przeprowadził Ramos
Pomoc w tłumaczeniu: Atreus & MrMat



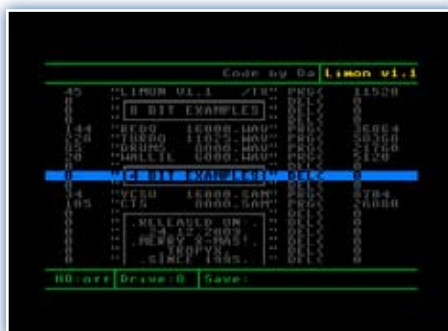
W tym dziale będziemy się starać zamieszczać zarówno nowe, jak i stare programy, które mało osób widziało, a które przyczyniały się do rozwoju zainteresowań ludzi komputerami C64 lub Amigą.

Pierwszy program pochodzi z czasopisma „Młody Technik” Nr 3/1986 – znalazł się on w dodatku specjalnym „Informik” w artykule o nazwie „*Ruchome obiekty na ekranie komputera Commodore 64*”. Autorami programu są Jacek Jędrzejowski i Mariusz Pietruszka.

Program wyświetla na ekranie monitora 80 obiektów, które szybko się poruszają. Każdy obiekt wykonuje ruch obrotowy. Można zmieniać szybkość ich przesuwania oraz wirowania. Program w prosty sposób demonstruje możliwości Commodore 64. Po jego uruchomieniu pojawia się pytanie, czy program mamy zachowany. Wpisujemy T i naciskamy RETURN. Uruchamiamy program wpisując sys49480, zatrzymujemy sys49482, natomiast szybkość uzyskujemy wpisując POKE 49323,a (gdzie zamiast a należy wpisać wartość od 1 do 5), animacja to POKE 49336,x (zamiast x wpisujemy wartość od 1 do 10).



LIMON V1.1

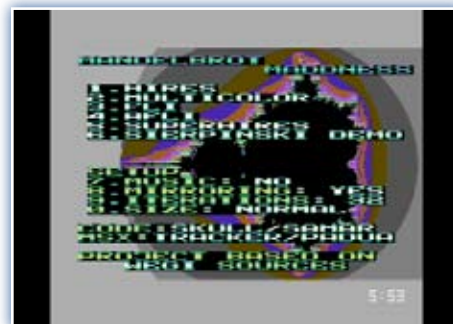


Rzadko się zdarza, aby w naszym kraju na C64 ktoś stworzył jakiś program użytkowy. Tym razem chcę Wam przedstawić program autorstwa Radosława Staszaka, znanego jako Data z grupy De-Koder/Tropyx. Program ten służy do konwersji 8-bitowych sampli do 4-bitowego formatu. Dzięki niemu możemy na naszym C64 uzyskać ciekawy sampling. Program wraz z przykładowymi plikami zajmuje prawie całą dyskietkę 5,25". Oprócz samego programu mamy cztery przykładowe 8-bitowe sample zapisane w pliku .wav oraz dwa 4-bitowe sample, które wcześniej zostały skonwertowane Limonem. Idea programu nie jest nowa, ale jego wykonanie bije dotychczasowe takie programy dla C64 na głowę. Sama konwersja z 8-bitowego na 4-bitowy sampling jest znacznie szybsza i prostsza. Mimo drobnych wad, program powinien znaleźć się w archiwum osób chcących tworzyć muzykę na samplach w pollytrackerze czy reflextrackerze lub wykorzystać jakieś wstawki sampli w swoich produktach.

Więcej informacji na temat programu znajdziecie na stronie: http://www.riversedge.pl/limon_v1_1_tropyx

MANDELBROT MADNESS

Program pokazujący jak długo tworzy się fraktale w różnych trybach graficznych C64. Jeśli ktoś posiada C128, to program korzysta z trybu Fast na tym komputerze. Powoduje to ponad dwukrotne przyspieszenie kreślenia fraktala (tryb AFLI w 4 min!). W wersji NTSC sam automatycznie dostosowuje się do wersji VIC'a i bez problemu działa na tej wersji komputera.



Wszystkie tu opisane programy znajdziecie na zamieszonej dyskietce C64 wraz z magazynem.

Ramos



SILESIAN AMIGA CLASSIC PARTY VOL. 5

**PO RAZ KOLEJNY ODBYŁO SIĘ SPO-
TKANIE AMIGOWEJ BRACI. TYM RAZEM,
GŁÓWNI ZE WZGLĘDU NA PANUJĄCĄ
NA ZEWNĄTRZ ZIMĘ, ZOSTAŁO PRZE-
NIESIONE DO GOŚCINNYCH POMIESZ-
CZEŃ MDK JORDAN W SIEMIANOWI-
CACH ŚLĄSKICH.**



Spotkanie odbyło się 23 stycznia. Jego for-
muła została rozszerzona i na meetingu oprócz
sporej rzeszy użytkowników klasycznych
Amig pojawili się także użytkownicy sprzętu,
który umożliwia uruchomienie MorphOS'a.
**Organizatorami spotkania byli Kamil i Radek
znani odpowiednio jako *y i Azzorek.** Wstęp
był w zasadzie bezpłatny. Niewielkie opłaty
wnosili tylko ci uczestnicy, którzy chcieli ko-
rzystać z dostępu do prądu. Podobnie jak przy
poprzednich meetingach, informacja o dacie i
miejscu spotkania pojawiła się z odpowiednim
wyprzedzeniem na portalu www.ppa.pl.

Użytkownicy jak zwykle nie zawiedli i
przywieźli ze sobą sporą ilość różnego amigo-
wego sprzętu. Dobrze, że obeszło się bez pro-
blemów z zasilaniem. Oprócz Amig były rów-
nież Macintosh'e, Efika oraz Commodore 64.

Jak zwykle na imprezach tego typu, uczest-
nicy po rozłożeniu swego sprzętu przystąpili
do łamania joysticków i testowania swoich
umiejętności w różnych grach – najczęściej
zręcznościowych. Królowały strzelanki, jak
stary dobry Project-X i ścigałki, jak Lotus.
Jednak największą furorę robił Mortal Komb
2. Stanowisko z tą grą było najbardziej oblega-
ne i dostarczało zarówno grającym, jak i ob-
serwującym najwięcej emocji. Tym razem na
złot przyjechała naprawdę duża ilość ludzi. Był
to wynik połączenia *SACP vol.5* i *krakowskich*

Amizaduszek, które wyjątkowo w 2009r nie
mogły się odbyć.

Większość osób zgromadzonych na sali
grała w różne gry, a Ci, którzy nie grali, gro-
madzili się w grupach i dyskutowali na naj-
różniejsze około-amigowe tematy i nie tylko.
Zawieranie znajomości trwało w najlepsze.
Miejmy nadzieję, że przyczynią się do po-
wstania w przyszłości kilku ciekawych pro-
jektów dla Amigi.

Jak już wspomniałem, oprócz użytkow-
ników Amig klasycznych tym razem przybyli
również *użytkownicy systemu MorphOS*. Dzię-
ki temu ludzie mający na co dzień dostęp do
Workbench'a mogli zapoznać się z interfacem
oraz zaletami jednego z alternatywnych syste-
mów, czerpiących z idei AmigaOS'u. Prezentacji
dokonywał **na swoim Macu Mini Pan Piotr
Waligórski**, znany użytkownikom Polskiego
Portalu Amigowego (PPA) jako wali7. Oprócz
kilku innych programów wali7 zaprezentował
także Stellarium – bardzo dobry program trak-
tujący o astronomii. Po dokładny opis odsy-
łam do pierwszego wydania Polskiego Pisma
Amigowego, w którym znajduje się recenzja
tego programu. Dodatkowo autor prezentacji
demonstrował możliwości MorphOS'a pod ką-
tem emulacji starych konsol i Amigi klasycz-
nej. Pokaz ten wzbudził dużą ciekawość wśród
uczestników zlotu. Wielu z nich wyraziło chęć
używania tego ciekawego systemu do chwili...
gdy poznali cenę oraz warunki licencji. No cóż,

150 euro i związanie posiadanej kopii z kon-
kretnym sprzętem odstraszyło wielu ludzi. A
szkoda, bo jest to ze wszech miar ciekawa al-
ternatywa dla współczesnych systemów opera-
cyjnych. Niestety, jesteśmy krajem na dorobku
i cena ma dla nas jeszcze decydujące znaczenie.
W tym miejscu warto również wspomnieć
o wizycie jednego z członków MOS Team'u.
Przybliżył on szczegóły nowej wersji systemu,
nad którą obecnie trwają wytężone prace. Trzy-
mamy kciuki – oby jak najszybciej się ukazała –
i apelujemy do użytkowników o jakąś recenzję
lub artykuł w stylu „Moje boje z...”.

OKOŁO GODZ 16 ODBYŁ SIĘ KON-
KURS NA MISTRZA GRY MORTAL KOM-
BAT 2. CAŁY TURNIEJ, A SZCZEGÓLNIE
FINAŁ DOSTARCZYŁ WSZYSTKIM NIE
LADA EMOCJI. ATMOSFERA BYŁA BAR-
DZO NAPIĘTA. ZACIĘTY POJEDYNEK
MIĘDZY AZZORKIEM, A ZNANYM Z
PPA BENEDYKTEM DZIUBAŁTOWSKIM
ZAKOŃCZYŁ SIĘ NIESPODZIEWANYM
ZWYCIĘSTWEM AZZORKA, KTÓREMU
GRATULUJEMY. BYŁO BARDZO WESO-
ŁO, A NA DOWÓD TRWAŁEJ PRZYJAŹNI,
OBAJ FINALIŚCI ZROBILI SOBIE ZDJĘCIE
Z „SZARFĄ” Z PAPIERU TOALETOWEGO Z
NAPISEM „BENEDYKT DZIUBAŁTOWSKI”.
AZZOREK ZABRAŁ „SZARFĘ” JAK CENNĄ
RELIKWIĘ I TROFEUM ZARAZEM, CHO-
WAJĄC JĄ W PRZEPASTNYCH CZELU-
ŚCIACH SVOJEJ BUDY :D. KTO BYŁ, TEN
WIE O CZYM PISZĘ.





WARTO WSPOMNIEĆ, ŻE TŁO IMPREZY WYPEŁNIAŁA MUZYKA Z GIER AMIGOWYCH, KTÓREJ MASTERING PRZYGOTOWAŁ TOBI. MIŁO BYŁO POSŁUCHAĆ ZNANYCH PRZEBÓJÓW W NOWYM „KLIMACIE”.

Późniejszy etap imprezy wypełniały rozmowy na najróżniejsze tematy, nie tylko Amigowe, oraz zabawy mniej lub bardziej znanymi grami. Podobno „nieczuły” na gry *y okazał się „czuły”, co dobitnie udowodnił, grając namiętnie w Wings of Fury :). Część uczestników

nadal próbowała swoich sił w grze Project X, jednak bez specjalnych sukcesów. Obowiązująca na imprezie pierwsza edycja tej gry okazała się znacznie trudniejsza od Second Edition z 1994 roku. Życzymy wszystkim twardzielowi więcej szczęścia w Project X podczas następnego spotkania SACP. Azzorek próbował swoich sił w grze Coala, jednak na oglądaniu intra się skończyło... Natomiast Alf i Doman89 zatracili się kompletnie w grze Moonstone. Silna motywacja do jej ukończenia została ostudzona po zabiciu harpii i utknięciu gry w tzw. „martwym punkcie” :D. Mordimer jeszcze przed turnie-

jem Mortal Kombat 2 próbował „roztrząsać” Azzorka, co sporadycznie mu się udawało, jednak „piesek w czarno-białe łatki” nie dał powodów do złudzeń, komu są dedykowane napisy „Finish Him!!!” :).



NO I W TYM MOMENCIE MÓJ OPIS SIĘ KOŃCZY. DALEJ AZZOREK ZAWIÓŻŁ NAS NA DWORZEC I TO, CO SIĘ DZIAŁO POTEM MUSI OPISAĆ ORGANIZATOR.

Podsumowując, meeting uważam za bardzo udany. Przybyło wiele osób, co zaowocowało nie tylko zacieśnieniem starych znajomości, ale również zawiązaniem nowych. Pozostaje pogratulować organizatorom świetnego pomysłu, organizacji oraz wyboru miejsca i czasu spotkania. Mam nadzieję, że za jakiś czas pojawi się informacja o organizacji kolejnego meetingu. Do zobaczenia!

RAPORT I FOTOGRAFIE: MRMAT





LISTY

Dziękuję wszystkim w imieniu swoim oraz całej redakcji FanCA – zarówno za dobre słowa, jak i za krytykę pod adresem magazynu. Wszystkie Wasze uwagi motywują nas do jeszcze lepszej pracy nad wydawaniem kolejnych numerów pisma. Zdziwilibyśmy się, gdyby nikt nie pisał o C&A Fan – to by nas bardziej zmartwiło, a tak mamy różne komentarze. Dzięki Wam za to! Nie wystrzegaliśmy się drobnych błędów, choć staraliśmy się dobrze przeżyć magazyn pod tym względem. Nikt nie jest doskonały i każdy może przeoczyć błędy.

Jeszcze raz muszę napisać, że magazyn jest projektem niekomercyjnym i hobbistycznym. Tworzymy go, poświęcając swój wolny czas i będziemy to robić, czy się to komuś podoba, czy nie. Chcemy Wam pokazać, że w pogoni za pieniądzem da się coś zrobić bezinteresownie, dla samej przyjemności tworzenia. Bardzo nas cieszy, że udało się nam skłonić choć jedną osobę (wierzymy, że więcej niż jedną) do wyciągnięcia z szafy komputera i stworzenia czegoś na nim lub po prostu zaciekawienia się nim. Zdumiało nas, że po C&A Fan sięgają ludzie nie związani ani z demosceną, ani z komputerami Commodore i dziwią się, że jeszcze ktoś interesuje się takim sprzętem i coś na nim tworzy. Z tego wynika, że nasza praca nie idzie na marne.

Na naszym redakcyjnym forum rozgorzała dyskusja o wersji angielskiej pisma. Pragnę wszystkich uprzedzić, że taka niestety nie powstanie. Ciężko znaleźć ludzi, którzy chcieliby pisać artykuły w języku Szekspira, nie mówiąc już o korektorach angielskiego. Inną kwestią jest sprawa, czy w angielskiej wersji magazynu miałyby się ukazywać zupełnie inne artykuły niż w polskiej, czy też tłumaczenia już napisanych. Łatwiej byłoby zapewnić to drugie rozwiązanie, ale niestety nie zawsze jest to możliwe. Bo jak przetłumaczyć komuś z Zachodu,

nie znającemu realiów komunizmu, pewne zwroty czy skojarzenia znane tylko Polakom?

Przejdźmy do odpowiedzi na Wasze pytania, bo trochę się ich nazbierało.

Czy jest szansa, aby pismo ukazywało się w wersji papierowej?

Ten temat już poruszaliśmy na łamach pisma (również na forum magazynu), ale muszę jeszcze raz napisać i wytłumaczyć, dlaczego magazyn nie może mieć wersji papierowej. Tu sprawa jest dość skomplikowana. Najważniejsza kwestia to zarejestrowanie pisma w sądzie i ustalenie statusu prawnego magazynu. Większość osób powie: olać to i wydawać pismo bez tych zezwoleń! Niby można i tak. Jednak nikt u nas w redakcji nie chce mieć najazdu Urzędu Skarbowego lub innego urzędu, któremu nie spodobałoby się, że wydajemy pismo nielegalnie i bez zezwolenia. Są też inne powody. Pismo papierowe musi ograniczyć liczbę artykułów i zmieścić się w wyznaczonej liczbie stron, aby nie przekroczyć ustalonej ceny za druk i wysyłkę. Z drugiej strony liczba chętnych do kupowania papierowego pisma jest mała w porównaniu do chętnych na zupełnie darmową wersję elektroniczną. To tym bardziej przekreśla możliwość wydawania pisma na papierze. Zresztą – przecież zawsze można samemu sobie wydrukować magazyn w domu na drukarce. Poza tym, powiedzmy sobie szczerze, komu się chce (albo raczej: kto ma czas) latać na pocztę i adresować kilkanaście przesyłek? Dlatego wersji papierowej póki co mówimy NIE. Ale w życiu bywa różnie, więc kto wie, może kiedyś takowa się pojawi.

Dlaczego nie ma opisu innych komputerów z rodziny Commodore?

Powód jest tylko jeden: nie ma kto się tym

zająć. To z kolei może wynikać z faktu, że w Polsce inne komputery niż C64/C128 i Amiga nie stały się bardzo popularne. Wiem, że sporo osób chciałby poczytać coś o tych komputerach, w miarę możliwości będziemy więc o nich pisać. Postaramy się znaleźć osoby znające się na takim sprzęcie i mogące coś ciekawego o nim powiedzieć na łamach pisma.

Czy pismo może być wydawane regularnie, np. co trzy miesiące?

Magazyn jako kwartalnik? Hmm.... Nie wiem, czy zmieścimy się w tym czasie i czy zawsze magazyn mógłby się tak ukazywać. Zależy to od wielu czynników, które muszą być spełnione. Jeśli w ciągu dwóch miesięcy powstanie wystarczająca ilość artykułów do złożenia pisma, to nie ma problemu. W przeciwnym przypadku termin się przeciągnie. Ważnym czynnikiem są też testy i korekta pisma, bo kilka osób zarzucało nam w ostatnim numerze pewne niedociągnięcia. Sama korekta zajmuje sporo czasu (niekiedy dłużej niż pisanie artykułu, co wynika z zapracowania korektorów), do tego dochodzą końcowe testy składanego pisma. Reasumując: zrobimy, co się da :)

Może znalazłyby się w piśmie kursy programowania, pisania muzyki czy obsługi programów użytkowych?

Takie kursy już powoli znajdują się w magazynie, ale ciągle szukamy ludzi do ich pisania. Nasz magazyn nie jest na razie pismem regularnym i cykl artykułów o programowaniu strasznie by się ciągnął, dlatego nie staramy się prowadzić takich kursów od podstaw, a po prostu zajmujemy się wybranymi zagadnieniami. W miarę naszych możliwości będą jednak powstawały tego typu artykuły.

Redakcja

WASZE OPINIE

Artykuł Skulla o NUFLI tak mnie wciągnął i zainteresował że zmusiłem się do sięgnięcia do źródeł czyli do obrazków w tym trybie i porównania teorii z praktyką i kodem w ciurku. Fajnie to zostało opisane i wytłumaczone miałbym zastrzeżenie tylko co do samego trybu że w określonych kolejnych 16 liniach rastra nie można zmienić jednocześnie kolorów 6 sprajtów a tylko 5 z nich albo kombinacje bo 6 cykli użyte na zmianę koloru jest wykorzystane na ustawienie wsp. Y dla ostatniej warstwy sprajtów.

Wydaje mi się że w 16-stu. Tylko w tym trybie jest tak, że pomijając sprajty to jedna linia jest krótka 23 cyklowa a druga długa 63 cyklowa. Uwzględniając 8 sprajtów to maksymalnie żrą one 19 cykli. Z tego co przeglądałem to rozumiem że właśnie 4 pozostałe cykle z 23 wystarczają na zrobienie STY \$d011 i wywołanie trybu FLI (albo inaczej, po wywołaniu STY \$d011 otrzymuje się krótka 23 cyklowa linia -19 cykli sprajtów i - 4 na STY i nie ma już czasu na żadne inne zmiany).

Zmiana kolorów następuje w następnej linii rastra, więc w dalszej części obrazka zamiast odwołania do jednego rejestru z kolorów sprajtów musi nastąpić ustawienie wsp.y sprajta. Z tego wychodzi że na ustawienie 8 duszków potrzeba 16 rastrów. W tym obszarze można zmienić maks 5 kolorów sprajtów z warstwy tych 6. Normalnie poza tymi trefnymi 16 liniami to jest tak. Zmiana kolorów sprajtów to 6*6 cykli= 36 cykli, plus 19 na sprajty to daje 55 cykli, z długiej 63 cyklówek linii zostaje 8 cykli. W tym czasie następuje ustawienie \$d018 (6 cykli) lub \$dd00 i zostają 2 cykle. W tych dwóch cyklach to już jest wartość dla \$d011 no chyba że są linie z STX \$d011. Bardzo fajnie

wyglądają obrazki samej bitmapy, sprajtów, AFLI, podziwiam że się chciało.

Fenek

No i jest dobry artek o grafice, można się dużo ciekawych rzeczy dowiedzieć - i chęć zaszczyć na Atari :)

Kaz

Moje pierwsze wrażenie, po przeczytaniu: „Łał, nie źle”. Z jednej strony, dobrze, że jest to PDF, bo można szybko zabrać się za czytanie (ściągasz, otwierasz i czytasz) Z drugiej jednak strony, źle się czyta na ekranie, choć klimat jest. Dobry art Atari vs Commodore. Co do opisu trybów graficznych... cóż, nie powaliło. Porównywalne możliwości z niewielką przewagą, wiadomo kogo]:->

Za to rozbawił mnie art „Muzyka bez SID'a” i tak przy okazji, kiedyś wpadł mi do głowy, pomysł szalony, by zagrać prostą melodię na... dysku twardym, ale jakoś szkoda mi było sprzętu który miałem, a na eksperymenty nie miałem „królika” :D Dziś dyski są za ciche, więc pomysł umarł. (to tak btw) Dobrym artykułem jest też: „Kiedy nie było neta”. Cóż, nie będę ukrywał, że „kącik programisty” i art o wektorach jest tym, co tygryski lubią najbardziej. Po za tym, świetny wstępniak ma ten art. W pełni popieram! no... koniec :)

Zilq

Przyznam że nie wiedziałem o tym że ktoś wydaje C/A . Format PDF mnie oczywiście nie przeszkadza a pomysł świetny .Przeczytałem ten numer i przyłączam się do opinii TDC dobra robota . Choć jeśli chodzi o konkretny artykuł Atari vs C64 mimo że autor niewątpliwie starał się podejść do tematu obiektywnie

wyczułem nutkę subiektywizmu no ale jest to pismo które adresowane jest do użytkowników Commodore :))

glowas11

Mimo wszystko jest to najbardziej obiektywny tekst o Atari vs C64 jaki kiedykolwiek powstał w Polsce ;) (a pamiętam co się działo w Bajtkowej redakcji C&A i Atari magazynu ;))

t4c

Kurdę, wywiad z Silver Dreamem jako pierwszy zaliczony (bo to legenda). Korzystało się z jego i Polonusa softu na C64. Świetnie się czytało. Lecimy dalej...

Deftronic/...

ojaja, jakość się podniosła o parę poprzeczek w porównaniu do początków , cud miód

rzookol

Poczytałem ... Dobra robota. Tylko screen Cytadali przy opisie Breathlesa daje po oczach. No i nie mogę się dopatrzyć końcówki jednego zdania w opisie kontrolera SCSI do A500. Ale w porównaniu do poprzednich - ten numer jest super.

MisOr

Jeśli ktoś ma ochotę napisać coś o Amidze do C&A Fan to zapraszam, chce bardziej więcej poruszać tematów związanych z tym komputerem. Jednak sam nie wiele zdziałam. Opisy gier nas już nie interesują, bo mamy ludzi od tego. Fajnie by było jakby ktoś zgłosił się od opisu programów, systemów itp. Może być też osoba, która przeprowadzi z kimś ciekawy wywiad ze świata Amigi. Z waszą pomocą magazyn szybciej się ukaże i będzie bardziej ciekawszy.

Ramos



AmigaONE
X1000