

# C&A fan

## AMIGA

- Historia Amigi cz. 8
- Sabrina 600 aka A600 Tower Mk II
- LHA cz. 3
- Gadżeciartwo i VisualPrefs cz. 1
- Perełki z AmiKIT - VirtualCD
- Wywiad z Lifter'em
- Polskie Pismo Amigowe 2/3

## Commodore 64

- Pograjmy jak za dawnych lat... 1984 cz. 2
- O przerwaniach i cyklowaniu na poważnie
- Commodore 64 GS
- Beeper czyli mimo, że soundchipa nie mamy to gramy!
- Wywiad z założycielem firmy BiW
- Silesia 4

# W NUMERZE

## WSTĘP

- 2 Od redakcji
- 3 News

## C64

- Pograjmy jak za dawnych lat... 1984 cz. 2 25
- O przerwaniach i cyklowaniu na poważnie 28
- Commodore 64 GS 37
- Beeper czyli mimo, że soundchipa nie mamy to gramy! 38
- Wywiad z założycielem firmy BiW 42
- Silesia 4 47

## AMIGA

- 4 Historia Amigi cz. 8
- 8 Sabrina 600 aka A600 Tower Mk II
- 9 LHA cz. 3
- 12 Gadżeciartwo i VisualPrefs cz. 1
- 17 Perelki z AmiKIT - VirtualCD
- 18 Wywiad z Lifter'em
- 22 Polskie Pismo Amigowe 2/3

## INNE

- Listy 50
- NUFLI Show 52

Tym razem mamy spore opóźnienie w wydaniu magazynu, ale najważniejsze że w końcu możemy oddać do waszych rąk najnowszy numer magazynu. W tym czasie w redakcji było trochę zawirowań i kłopotów, ale mimo wszystko dalej działamy i nie poddajemy się. Jeśli chcecie, aby magazyn ukazywał się w miarę regularnie to czekamy na wasze artykuły. Szukamy ludzi którym od czasu do czasu chciałoby się zasilić nasz magazyn lub stronę www swoim artykułem. Ze względu na małą ilość redaktorów, sami wszystkim nie jesteśmy w stanie się zająć.

Oprócz pracy nad magazynami staramy się również tworzyć materiały na stronę. Tam oprócz naszych i nadesłanych przez czytelników artykułów możecie poczytać garść newsów. Staramy się dawać takie newsy, które są mało znane i warte większej publikacji. Nie chcemy powtarzać i kopiować newsów z innych stron związanych z C64. Mile widziane są Wasze komentarze pod newsami czy artykułami oraz aktywny udział na naszym forum. Dzięki temu będziemy wiedzieć o czym pisać, co jest dla Was najważniejsze. Zapraszamy do uczestniczenia w życiu strony magazynu.

W tym numerze fani Amigi będą mogli poczytać kolejny odcinek historii swojego komputera. Tym razem widzianej oczyma twórców gier. Fani ulepszania biału mogą poczytać na temat możliwości VisualPrefs. Zapraszamy także do zapoznania się z ciekawym naszym zdaniem wywiadem.

Natomiast fani Commodore 64 poczytają o tym jak można odgrywać muzykę bez użycia SIDA i na czym taką muzykę tworzyć. Osoby uczące się programowania zaciekawi z pewnością artykuł o przerwaniach i cyklowaniu. Z pewnością wiele osób zainteresuje wywiad z założycielem firmy BiW, z którego możecie się dowiedzieć jak ta firma działała i co było powodem zakończenia działalności.

A co dzieje się w samej redakcji? Wielkich zmian nie było. Nadal piszemy artykuły i udzielamy się na stronie. Trwają prace nad ulepszeniem jej oraz forum. Proszę regularnie zaglądać i komentować wszelkie zmiany a my postaramy się je wszystkie uwzględnić (oczywiście w miarę możliwości).

Podjęliśmy także współpracę z redakcją magazynu KOMODA i od czasu do czasu na stronie będą się pojawiać artykuły ludzi związanych z tym pismem. Magazyn KOMODA można już pobierać z archiwum naszej strony. Najważniejsze adresy kontaktowe można znaleźć w stopce redakcyjnej obok. Czekamy na Wasze komentarze, podpowiedzi i artykuły.

Do następnego numeru.

**Redakcja**

E-Mail: [admin@ca-fan.pl](mailto:admin@ca-fan.pl) (w sprawie strony)  
[ramos@ca-fan.pl](mailto:ramos@ca-fan.pl) (w sprawie magazynu)  
Strona: <http://ca-fan.pl/>  
Nasza-Klasa: <http://nk.pl/#grupy/84536>  
Facebook: <http://www.facebook.com/pages/Commodore-Amiga-Fan/14489460228504?v=wall>

## Commodore & Amiga Fan

Magazyn użytkowników komputerów firmy Commodore

**Redaktor naczelny:** Ramos

**Autorzy artykułów:** arti, Don Rafito, MrMat, Nitro, p.a., Ramos, Skull.

**Założa FanCA:** ti, Atreus, Bago Zonde, Black Light, Don Rafito, Indyjr, Milek, MrMat, p.a., Ramos, Skull, TOUDIdel.

**Korekta:** arti, Atreus

**Fotografia reportażowa z Silesia 4:** Randy

**Fotografia na okładce oraz w temacie numeru:** Wikipedia, Christian Rosentretnetner

**Desing, skład i łamanie:** Black Light - procreators.pl

# Newsy Newsy Newsy



## CheeseCutter

Pojawiła się wersja beta edytora muzycznego. Jest to port znanego z C64 edytora muzycznego o nazwie JCH, napisany na PC. Edytor jest rozwijany pod licencją GNU GPL przez Timo Taipalus znanego jako Abaddon / Fairlight. Obsługuje wszystkie popularne playery napisane przez JCH, w tym 20.G4. Pomimo wersji beta 0.3.21, edytor działa bardzo stabilnie i poprawnie odgrywa muzykę z C64. Razem z edytorem można ściągnąć paczkę pod nazwą 'Worktunes' z ponad setką kompozycji między innymi Draxa, Metala, M&D i kilku innych znanych muzyków z C64.

<http://koti.kapsi.fi/~ttaipalus/ccutter/>

## Independent Zin

Oprócz magazynu PPA czy KOMODA pojawił się nowy magazyn o nazwie Independent Zin. Oto jak pisze redaktor naczelny magazynu, lukasz050792: „Independent Zin - to magazyn o wszystkim co oldschoolowe, znajdziesz w nim m.in. recenzje starych gier, filmów a także felietony, poradniki, kody, dodatki, screen'y i masę innych rzeczy związanych z starymi dobrymi czasami.” Magazyn jest wydawany w pliku html regularnie co miesiąc 10-tego.

[http://stare.e-gry.net/independent\\_zin](http://stare.e-gry.net/independent_zin)

## TurboDemoCard

Jest to następny produkt znanego elektronika Kisiela. Tym razem stworzył on

dopalkę do C64 opartą na podkręceniu płyty głównej do 2Mhz. Jest to bardzo ciekawy projekt, bo dzięki temu większość rzeczy chodzi o wiele szybciej i daje to nowe możliwości C64. Więcej o projekcie na stronie: <http://projekt64.filety.net/index.php?dir=TDC/>

## CBM prg Studio

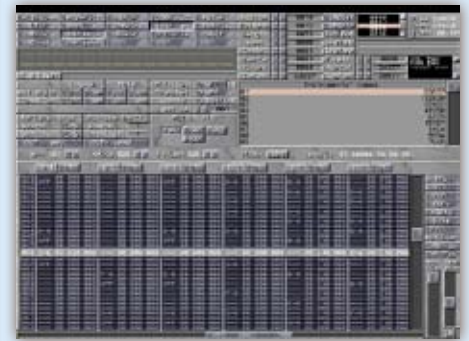
CBM prg Studio, to program stworzony na Windowsa. Służy on do pisania programów w BASICU C64 lub VIC20, które później można zapisać jako plik prg i uruchomić na komputerze lub emulatorze. Niestety na obecną chwilę projekt jest jeszcze w wersji rozwojowej. Program ściągamy ze strony:

<http://www.ajordison.co.uk/>

## No Pets Allowed

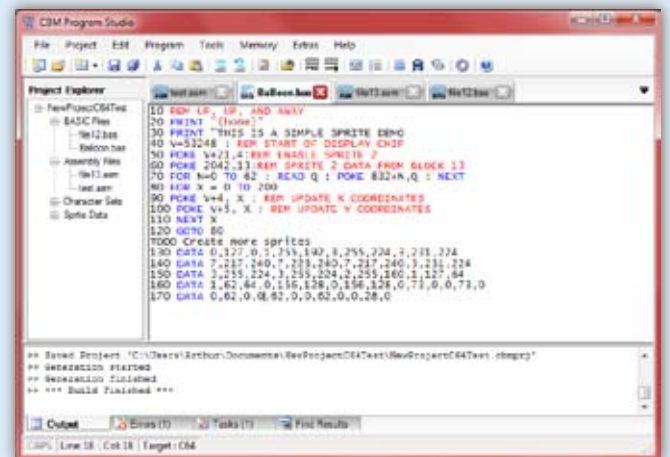
Kto by pomyślał, że powstanie pierwsze z prawdziwego zdarzenia demo na komputer PET. Do tej pory nikt nie chciał pokazać możliwości jakie drzemią w tym prostym, pierwszym komputerze firmy Commodore. No Pets Allowed to produkcja grupy ORB, która znana jest z podobnych produkcji na VIC-20. Za demo odpowiadają: Ultra (code), 4mat (code and music) oraz Ilkke (graphics).

Demo można zobaczyć: <http://www.youtube.com/watch?v=ktUguF1TnJk> lub ściągnąć z: <http://www.pouet.net/prod.php?which=56410>



## DIGIBOOSTER Pro

Jeden z najbardziej znanych edytorów muzycznych, który powstał na AmigaOS ukaże się jeszcze w tym roku. Z oficjalnej strony programu można pobrać wersje demonstracyjne skompilowane dla AmigaOS 4.x oraz MorphOS-a. Udostępniony program jest tylko wersją demonstracyjną i braku mu wielu opcji. Więcej można poczytać na: <http://www.digiboosterpro.de/>



## Podsumowanie 2010

Koniec starego roku to podsumowanie tego co się wydarzyło i tak na kilku stronach ukazały się podsumowania 2010.

Co się wydarzyło na Amidze zapraszam: [http://exec.pl/artykuly/podsumowanie2010/amigowe\\_podsumowanie\\_roku\\_2010.html](http://exec.pl/artykuly/podsumowanie2010/amigowe_podsumowanie_roku_2010.html)

Najlepsze gry 2010 na C64 według Akiry: <http://www.scenial.pl/archives/4100>

oraz Scena 2010: <http://www.scenial.pl/archives/2438>

# HISTORIA AMIGI cz. 8:

## Cień 16-bitowej Bestii: rzut okiem wstecz na amigowe granie

Notatka autora:

Chcę osobiście podziękować dosłownie setkom ludzi, którzy odpowiedzieli na moją prośbę o historię od programistów gier na Amigę, bez których ten artykuł nie mógłby powstać. Niestety, nie było możliwe, aby wszystkie te opowieści zamieścić w artykule, ale uczciwie spróbowałem odpowiedzieć na każdego emaila, którego otrzymałem. Jeżeli kogoś pominąłem, przepraszam.

### Wprowadzenie



Amiga narodziła się jako maszyna do gier, ale pojawiła się w świecie, w którym przemysł gier wideo był już dobrze ugruntowany i zmieniał się szybko. Dawno minęły czasy, kiedy samotny koder całymi nocami przesiadywał w piwnicy przez sześć tygodni i wyskakiwał z jakimś hitem na Atari 2600. Nawet młodszy i mniejszy przemysł gier komputerowych dawno już minął etap, w którym Roberta Williams pakowała dyskietki do woreczków foliowych i odbierała telefony od graczy w swojej kuchni. Sukces Commodore 64 (a po drugiej stronie oceanu, sukces Sinclair Spectrum) oznaczał większe pieniądze na programowanie gier komputerowych. Było to całkiem dobrą sprawą w momencie, gdy potężniejsze 16-bitowe maszyny zaczynały poważnie wskazywać ograniczenia jednoosobowej drużyny programistycznej.

Po raz pierwszy w branży produkcji gier pojawiały się wyspecjalizowane zawody. Bogata, 4096-kolorowa paleta barw Amigi do tworzenia duszków i też wymagała ludzi uzdolnionych artystycznie. Czterokanałowy sampleowany chip dźwiękowy wręcz wołał o muzyków, którzy sprawiliby, że zaśpiewa. Większy rozmiar i złożoność gier wymagały, aby ktoś inny

niż programiści testował gry przed ich wydaniem. I wreszcie konieczne były nowe stanowiska zarządzania, by nadzorować pracę tych twórczych ludzi.

### Życie w okopach

Znalezienie tych ludzi nie było łatwe. W wielu przypadkach była to kwestia osób nie-doświadczonych, zatrudniających swoich nastoletnich rówieśników – tych, których znali z liceów czy klubów komputerowych. Klika większych firm deweloperskich, takich jak Ocean, posiadało pulę swoich wewnętrznych programistów, ale dla większości gier prace były zlecane zewnętrznemu zespołom. Zespoły te, często obsadzone przez zupełnie zielonych programistów, były niebezpiecznie niepewne. Większość z nich nigdy nie zrobiła więcej niż dwóch gier. Zderzające się ego i argumenty, walka o ubogo sformułowane albo nieistniejące kontrakty i znikające fundusze naciągały przyjaźnie do granic wytrzymałości. Kiedy studio desperacko potrzebowało gotówki, programiści i artyści pracowali po 16 i więcej godzin dziennie. Menedżerowie wydzwaniali do swoich pracowników kontraktowych o drugiej nad ranem tylko po to, by się upewnić, że wciąż pracują. Miha Rinne, który pracował dla Terramarque, opowiedział mi, jak jego kierownik zażądał, by ten szczegółowo opisał cały czas, jaki spędzał robiąc notatki, wykonując kopie danych czy nawet idąc do toalety! Później wykorzystał swoje doświadczenia w przemyśle gier wideo do stworzenia komiksu, który można znaleźć tutaj: <http://peliala.wordpress.com/2008/10/11/hello-world/>

Pomimo długich godzin pracy i niskich wynagrodzeń, wciąż było mnóstwo ludzi chętnych, by podjąć się tworzenia gier na Amigę i wielu takich, którzy uważali się za szczęśliwców, iż są częścią tego wszystkiego. Daniel Filner, który przeprowadził na Amigę klasyczne tytuły LucasArts, takie jak LOOM czy Indiana Jones, powiedział mi: „Kiedy miałem 17 czy 18 lat, na liście ‘praca moich marzeń’ było coś takiego: 1) superbohater, 2) pilot X-Winga lub jakiś inny rodzaj astronauty, 3) programista gier wideo”.

Zarówno wtedy, jak i teraz istniał podział pomiędzy programistami a wydawcami i cele tych dwóch grup nie zawsze były zbieżne. Wydawcy chcieli zmaksymalizować dochód, tak więc wiele gier wypuszczano na „wszystkie formaty” – Amigę, Atari ST, zubożoną wersję na IBM PC, były nawet wersje na starsze komputery 8-bitowe, takie jak Commodore 64 czy Sinclair Spectrum. Wymaganie, by gry zmierzały w stronę najmniejszego wspólnego mianownika, ograniczało programistów, którzy chcieli wykorzystać całą moc Amigi.

Tłumaczenie poniższego komiksu:

Obrazek 1: A: Co powiedziałeś?? B: Mówię, że muszę dostać trochę pieniędzy z góry, bo inaczej nie będę mógł przyjeżdżać do pracy!

Obrazek 2: A: Więc mówisz, że musimy ci zapłacić teraz albo nie będziesz pracował? Nie lubię, jak mi się grozi! B: To nie była groźba!

Obrazek 3: C: Proszę posłuchać, jemu chodziło o to, że ma tak mało pieniędzy, że nie stać go na bilet kolejowy.)



Blaski pracy programisty gier

Niektóre firmy programistyczne, takie jak Psygnosis, poszły inną drogą. Skoncentrowały się na maszynach 16-bitowych i próbowały osiągnąć tak wielki efekt graficzny, jak tylko mogły. Taka strategia okazała się sukcesem, ponieważ właściciele starszych maszyn mieli teraz nieodparty powód, by ulepszyć swój sprzęt. Kiedy minęły lata 1980-te, inne firmy poszły za tym przykładem. Amiga stała się pokazową maszyną dla gier, poszerzającą granice tego, co uważane było za możliwe. To z kolei przyciągało wyjątkowych ludzi, takich jak artysta Jim Sachs, którzy zapalili się do pomysłu stworzenia czegoś nowego.

„Naprawdę cieszyły mnie te wczesne dni Amigi” – mówił Jim. – „Nie mogłem się doczekać, kiedy wstanę rano, wiedząc, że będę tworzył całkiem nowe efekty, jakich nikt wcześniej na ekranie komputera nie widział”.

Mimo że scena gier komputerowych była – podobnie jak jest teraz – dużo mniejsza niż rynek gier konsolowych, miała ona swój wpływ na szerszy świat. Michael Crick, autor gry WordZap, pamiętał historię, kiedy to ówczesny dyrektor naczelny Nintendo (którego córka przyjaźniła się z córką Michaela) zastał kiedyś Michaela grającego w *Defender of the Crown* na swojej Amidze. Szef Nintendo, którego firma niczym kolos rządziła wówczas światem gier video, był tylko w stanie gapić się na maszynę oniemiały i mamrotać raz po raz o „wspaniałej grafice”.



Grafika Jima Sachsa do gry *Defender of the Crown*

## Magazyny i recenzenci

Z tego przemysłu urosła cała społeczność. Istniejące i dopiero zakładane czasopisma o grach stały się punktem skupiającym wokół siebie programistów, ich gry i użytkowników. Tom Malcom spędził sześć lat recenzując gry na Amigę dla *.info*, jednego z najpopularniejszych magazynów w tamtych czasach. „Jeśli była jakaś gra na Amigę, prawdopodobnie ją widziałem i w nią grałem” – mówił Tom. – „Każdego dnia ciężarówka UPS, zawsze nazywana Ciężarówką Z Zabawkami, podjeżdżała do tylnego wyjścia i zostawiała kolejny ładunek gier”. W niektóre dni przybywało ponad tuzin nowych tytułów.

Związki pomiędzy recenzentami a programistami były przyjacielskie i bliskie. Były to jeszcze niewinne czasy, przed dniami natrętnych firm PR i naciskami na recenzentów, by dostarczali oni „odpowiednich” ocen. Tom często odwiedzał biura Psygnosis. Jego ulubioną grą była mało znana przewijana strzelanka Menace, do której programiści zdradzili mu kod: „xr4titurbonutterbastard”. Kod ten oparty był na samochodzie i umiejętnościach kierowania jednego z programistów (*turbonutterbastard* – *turboświrowydrań* – przyp. tłum.), z którym reszta zespołu obawiała się jeździć. Później, podczas pokazu handlowego w Chicago, Malcom wziął kilku programistów Psygnosis na przejażdżkę swoim kabrioletem, co było sposobem na odwzajemnienie się.

W październiku 1989 Tom miał jechać do Kalifornii, by odwiedzić kilku wydawców gier. Trzy dni przed jego wyjazdem miało miejsce trzęsienie ziemi Loma Prieta. „Sufity na lotnisku w San Francisco leżały w stosach na podłodze, nabrzeże wciąż było w ogniu, fragment mostu Bay Bridge był złamany, a autostrada w Oakland została całkowicie spłaszczona” – opowiadał mi, przywołując tamto surrealistyczne doświadczenie. „Odwiędziłem Electronic Arts w Redwood City, gdzie David Dempsey, jeden z ludzi EA od marketingu, pokazał mi ogromne pęknięcia na klatkach schodowych. Ludzie byli trochę niespokojni, ale twardo próbowali się trzymać. Byłem trochę zdenerwowany tym, że miano mnie przedstawić tym ludziom tak krótko po trzęsieniu, ale oni wydawali się zadowoleni z faktu, że coś odwraca ich uwagę od tych przykrych zdarzeń”. Nawet trzęsienie ziemi nie wystarczyło, by powstrzymać tych programistów od ich pracy.

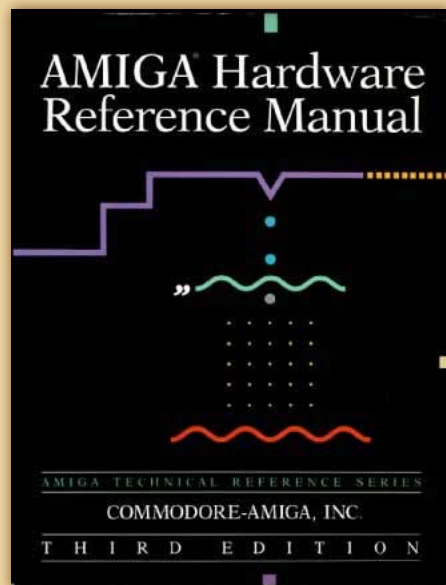
Kiedy gry były już zaprogramowane i recenzowane w magazynach, następnym krokiem w ich podróży do klienta był sklep. W tamtych czasach, większość miejsc, w których sprzedawano gry komputerowe to były niezależne firmy, z których każda miała swój własny wygląd i osobowość. Kevin Hollingshead prowadził oddział Program Store, będący jednym z pierwszych sklepów, które stworzyły sieć. Sprzedawali gry na Amigę, Atari ST i na inne platformy. Pewnego popołudnia Trip Hawkins (założyciel Electronic Arts) pokazał się w jego sklepie po tym, jak okazało się, że ten oddział sprzedaje więcej gier EA, niż jakiegokolwiek inny punkt sprzedaży detalicznej. Trip opowiedział Kevinowi o swoich ulubionych grach oraz o swoim głównym planie, by zrobić z Amigi jeszcze większy sukces, usprawiedliwiający wszystkie inwestycje i entuzjazm, który on osobiście włożył w tę platformę. Zaplanował próbne porozumienia z japońskimi spółkami, aby stworzyć konsole do gier oparte na chipsecie Amigi, a także „Amigę-na-karcie” dla IBM PC. Trudno przewidzieć, co stałoby

się ze sceną gier na Amigę, gdyby te plany się ziszczyły, ale koniec końców nie miało to znaczenia, gdyż Commodore ukręcił obydwie pomysły. Mimo wszystko przemysł związany z grami na Amigę wciąż prosperował, gdy lata 80-te przeszły w 90-te. W miarę jak programiści dowiadywali się więcej na temat tej platformy, Amiga stawała się znana jako komputer z najlepszymi grami. Wydawcy często umieszczali amigowe zrzuty ekranowe z tyłu pudełek z grami, nawet w wersjach na Atari, Commodore 64 i IBM PC, na których poniżej drukowano małą napis: „Pokazano wersję na Amigę”.

## Odkrywanie tajemnic programowania gier

Sztuczek ani technik, wymaganych do robienia świetnych gier na Amigę nie uczono w żadnych szkołach. Programiści często zaczynali od zabawy w wbudowanym językiem BASIC, jak w Commodore 64, a potem próbowali bawić się z assemblerem. Zanim wypuszczono na rynek Amigę, magazyny takie jak Commodore Gazette czy MC MicroComputer zawierały artykuły, które traktowały o wnętrznościach sprzętu.

Mimo to jedynym liczącym się podręcznikiem, który mógł pomóc naprawdę zrozumieć moc chipsetu Amigi, był commodorowski Amiga Hardware Reference Manual (poradnik dotyczący sprzętu Amigi). Była to biblia programistów amigowych gier. Pozwalał oddanym badaczom odkryć, w jaki sposób blitter umieszczał grafikę bezpośrednio z pamięci na ekranie, jak copper pozwalał programiście wskoczyć i zmienić sposób wyświetlania nawet w środku przeglądania linii ekranu, jak chip audio zajmował się przetwarzaniem dźwięku, czy wreszcie jak procesor synchronizował wszystkie te działania.



Biblia programisty

Pierwszą rzeczą, jaką robili programiści przystępując do pisania gry, było delikatne odcięcie systemu operacyjnego, by zyskać całkowitą kontrolę na sprzętem, włącznie z pamięcią. Robiono to, by zaoszczędzić pamięć, żeby gry mogły wyciskać całą dostępną moc z dedykowanych chipów. Jednakże kiedy gra była uruchamiana, programista nie dysponował żadnymi zabezpieczeniami. Nie było żadnego zintegrowanego środowiska programistycznego albo debuggera, który mógł działać na tym etapie. Sprytni programiści używali samego sprzętu jako monitora, wysyłając sygnały do *coppera*, aby zmieniał kolor tła w kluczowych momentach w programie.

Kiedy coś szło źle, to szło źle bardzo widowskowo. Nie było żadnej ochrony pamięci, więc jeśli jedna część programu zaczynała nadpisywać inną, ekran pokrywał się przypadkowo generowanymi fajerwerkami, zanim system się zawiesił. Jedynym sposobem, by upewnić się, że całość będzie działać, było budowanie i testowanie określonych procedur dla każdego elementu gry, zanim zaczęło się prace nad kolejnym.

Programiści gier na Amigę stosowali w swojej pracy wiele różnych narzędzi. Aby oszczędzić czas spędzany na programowaniu, używano wielu języków wysokiego poziomu, głównie C. Popularnymi kompilatorami były Lattice C firmy SAS, Manx C i DICE. Jeżeli chodzi o szybkość i moc, assembler 68000 wciąż był nie do pobicia. We wczesnych Amigach, pamięć była na wagę złota: rynek składał się z maszyn takich jak Amiga 500 z 512 KB (kilobajtami, nie megabajtami!) RAM, dzielonymi pomiędzy pamięcią „chip” (pamięć dedykowana wyświetlaniu i specjalistycznym chipom) i „fast”. Sprawienie, że wszystko będzie działać gładko, bez migotania i przy wysokich szybkościach klatek, wymagało niezłej umysłowej żonglerki.

Pomyślcie tylko o zadaniu Cesare Di Mauro, programisty USA Racing. Chciał osiągnąć 50 klatek na sekundę, by zachować płynność, ale biorąc pod uwagę ograniczenia RAM, nie było możliwe użycie podwójnego buforowania (jest to technika, w której drugi ekran z kolejną klatką jest przechowywany w pamięci). Użycie pojedynczego ekranu oszczędzało czas wykonania i pamięć, ale znacznie skomplikowało zadanie przewijania tła i uaktualnianie BOB-ów (obiektów blittera, podobnych do duszków). Jego rozwiązanie polegało na użyciu wirtualnego ekranu 352 x 272, przy czym tylko 320 x 240 pikseli było widocznych. Obszar podzielony został na dwa pionowe wycinki, połączone tak, by pokazywały pojedynczy widok. Tło składało się z płytek 32 x 32 pikseli, zamieszczonych na wielkiej mapie 4096 x 65536 pikseli (koderzy

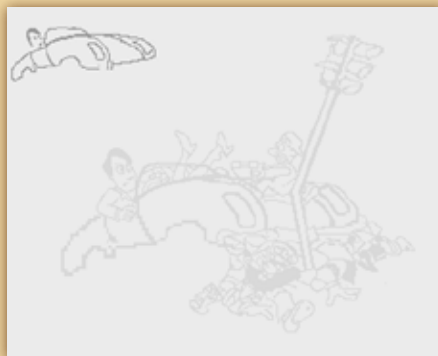
wszędzie będą w stanie rozpoznać te liczby).

Żonglowanie wieloma płytkami, wyświetlanymi na wierzchu BOB-ami, muzyką i efektami dźwiękowymi oraz wykrywaniem kolizji ze ścianami i innymi samochodami – było gigantycznym wyzwaniem. Cesare w końcu napisał narzędzie w assemblerze, które radziło sobie z tym wszystkim przy 50 klatkach na sekundę, sortując obiekty do narysowania na liście wyświetlania, w celu zwiększenia wydajności. Program najpierw wyświetlał w porządku wszystkie BOB-y na liście, następnie aktualizował ukryte obszary, by zająć się przewijaniem (przewijanie poziome było realizowane przez chipset graficzny poprzez ustawienie wartości scrollu, podczas gdy przewijanie pionowe było podobne, lecz bardziej skomplikowane), potem czekał aż promień wyświetlania monitora osiągnął dolną pozycję każdego BOB-a, by przywrócić tło, które „poplamili”. Napisał specjalne procedury dla dźwięku, a nawet dla dostępu do dysku, by zmaksymalizować szybkość i zminimalizować zużycie RAM. Ten rodzaj ostrożnego balansowania był typowy dla programistów gier, którzy chcieli przekraczać bariery. Wiele pomysłów, na które wpadli zostało znacznie później odtworzonych w standardowych bibliotekach dla tego przemysłu – listy wyświetlania, na przykład, są kluczowym elementem DirectX3D.

## Malowanie na płótnie

Oddani graficy wcale nie byli mniej innowacyjni. Deluxe Paint firmy Electronic Arts był głównym programem graficznym w tamtym czasie, ale nie miał wielu przydatnych funkcji (takich jak warstwy czy historia), które Photoshop później uczynił standardem. W tamtych czasach tablety graficzne i skanery były bardzo drogie, więc wielu artystów musiało improwizować.

Miha Rine pracował nad wieloma obrazami dla amigowych gier. W poniższym przykładzie miał stworzyć obraz tła dla gry wyścigowej. Jak zwykle czas był bardzo istotny. W typowym toku pracy Miha zacząłby od narysowania myszką szkicu konturowego przy 2-krotnym zoomie i używając najjaśniejszego gradientu.



Zakończył konturowanie używając ciemniejszego odcienia. Kiedy był już pewny wyglądu obrazka, mógł przystąpić do kolejnego etapu: kolorowania.



Aby zaoszczędzić pamięć, powszechnie używano trybu 32- albo nawet 16-kolorowego. Przez to niezwykle istotne było dobranie właściwej palety kolorów. Omawiany obrazek był pierwszym dziełem tego artysty w 256 kolorach.

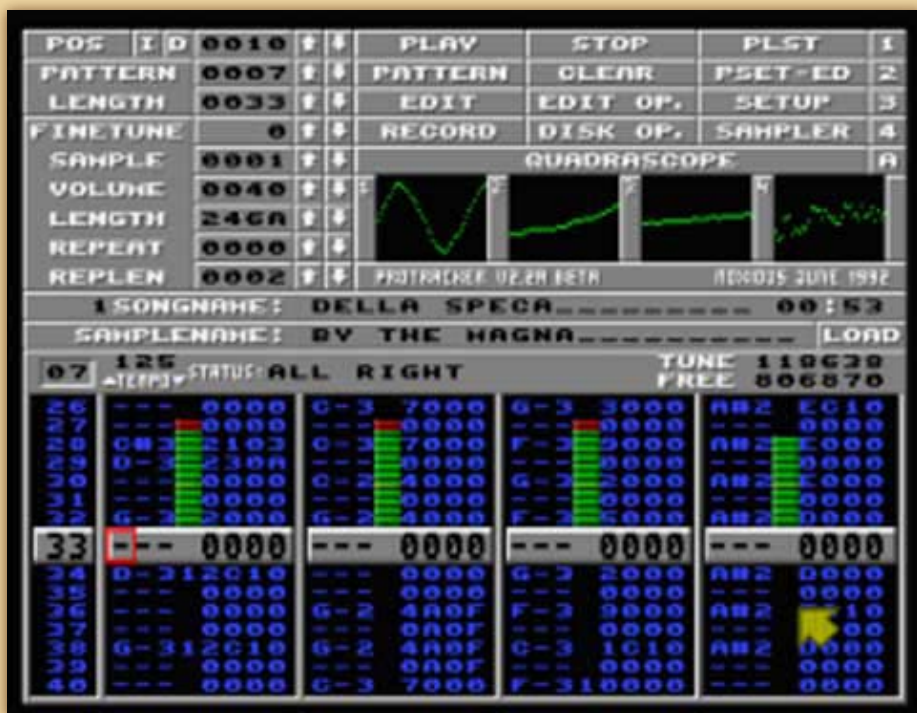


Następny etap polegał na dodaniu cieniowania i wygładzania krawędzi. Miha zrobił to ręcznie, piksel po pikselu. Powoli oryginalny kontur zanikał.



Końcowy obraz. Było to ostatnie dzieło, jakie Miha stworzył dla firmy, w której wówczas pracował.

Czasami prosty efekt mógł zrobić w grze ogromną różnicę. Christopher Jackson, programista amigowej wersji *Wayne Gretzky Hockey 3*, dodał animację hokeisty wbijającego krążek w kierunku gracza i „roztrząskującego” monitor w momencie, kiedy krążek sięgnął celu. Pojawiły się też inne smaczki: małe pikselowe ślady za graczami, które Zamboni\* mógł później wyrównać. Nie dziwi więc, że inne wersje



Zrzut ekranowy programu Protracker

Wayne Gretzky Hockey miały na pudełkach zrzuty ekranowe z wersji dla Amigi.

Wielu artystów związanych z grami, którzy przeszli na Amigę poczuło to zdumiewające poczucie wolności, którego wcześniej nie doświadczali. Manfred Kramer, późniejszy artysta-grafik komputerowy, pracujący nad grami wideo i filmami (ostatnio nad Avatarem), napisał:

„Używałem C64 do robienia grafiki pikselowej, możecie więc sobie wyobrazić, jak się poczułem, gdy po raz pierwszy miałem prawdziwą paletę kolorów i myszkę! Dokładnie pamiętam, co czułem, kiedy poruszyłem myszką i dodałem żółty kolor do ekranu w czasie rzeczywistym. Pamiętam też, że miałem łzy w oczach, kiedy zobaczyłem jak moja pierwsza scena wyrenderowana w 3D pojawia się na nowonabytym framebufferze dla Amigi 2000.”

## Tworzenie dźwięku

Nie mniej ważne niż pisanie kodu czy tworzenie grafiki, było uzupełnianie ścieżki dźwiękowej. Nawet oszałamiające graficznie gry, takie jak Shadow of the Beast, znane były z powodu ich niezwykle poruszającej muzyki. Nie ograniczeni już do pisków i szumów, muzycy związani z Amigą mogli bawić się cztero-kanałowym smpłowanym dźwiękiem stereo. Pliki muzyczne Amigi, nazywane MOD-ami, przechowywały próbki każdego instrumentu wraz z informacją na temat czasu trwania, intensywności i efektów specjalnych, stosowanych do każdej nuty. Pliki te tworzone były za

pomocą programów zwanych sekwenserami, które obejmowały aplikacje freeware, takie jak Noisetraacker czy Protracker, aż po specjalne kodowane sekwensery.

Tworzenie muzyki dla gier na Amigę było zazwyczaj szybkim zadaniem, więc muzycy byli wprowadzani zazwyczaj pod koniec projektu, na ogół na okres kilku tygodni. Matt Simmonds musiał raz pracować przez weekend, ponieważ konieczne było stworzenie ścieżek dźwiękowych do dwóch gier już na poniedziałek. W ciągu dwóch dni napisał około 30 piosenek – poziom produktywności nieczęsto widziany nawet dzisiaj.

## Nigdy się nie poddać

Techniki twórczego programowania, całkiem nowe efekty graficzne i innowacyjna muzyka były typowe dla twórców gier na Amigę. Ale kim byli ci ludzie? Co nimi kierowało? Jim Sachs ma odpowiedź:

„(...) Nawet wtedy tak naprawdę nie chodziło o sprzęt. Chodziło o specyficzny typ ludzi, których ten sprzęt przyciągał, z ich podejściem typu 'to się da zrobić'. Kiedy po Amidze zacząłem konsultować projekty związane z PC, byłem zdumiony, że programiści nie garnęli się, by czegoś spróbować, jeżeli jakiś inny programista nie zrobił tego wcześniej. W przypadku Amigi niemal miało się z celem próbowanie jakiegoś efektu, chyba że NIKT jeszcze go nie osiągnął”.

Ten rodzaj ludzi żył na całym świecie, niekiedy w bardzo trudnych okolicznościach.

Rabah Shihab był studentem Uniwersytetu Bagdadzkiego w Iraku we wczesnych latach 1990-tych. On oraz jego przyjaciele: artysta Murtadha Salman oraz muzyk Mahir AlSalman, wspólnie zaprogramowali amigową grę Babylonian Twins. Gra była stworzona na komputerze Amiga 500 z jedynie 512 KB RAM-u i miała imponującą grafikę, inspirowaną tekstami historycznymi. Mimo, że gra była niemal skończona, pierwsza wojna w Zatoce i późniejsze sankcje na Irak uniemożliwiły jej wydanie.

Jednakże Rabah nie poddał się. Dekady później, już mieszkając w Kanadzie, ocalił kluczowe elementy gry i ponownie zjednoczył swój stary zespół, by dokończyć produkt. Wersja demonstracyjna z oryginalnym kodem amigowym została wreszcie udostępniona publicznie, z zapowiedzią pełnego wydania. Jednakże na scenie właśnie pojawiła się kolejna platforma do gier, na której niewielkie grupy niezależnych programistów wciąż mogą konkurować z olbrzymimi spółkami. Rozszerzona wersja Babylonian Twins ostatnio pojawiła się na iPhone, a w wersji HD także na iPad. Grałem w tę grę, jest to absorbująca, przewijana bocznie platformówka z elementami logicznymi, coś w stylu skrzyżowania Prince of Persia z The Lost Vikings. Gra już zdobyła kilka nagród. Możecie pobrać ją przez Apple Store lub bezpośrednio ze strony <http://www.babyloniantwins.com>.



Babylonian Twins na iPhone

W kolejnym odcinku będę zgłębiał 'demo-scenę' – grupy artystów, muzyków i koderów, którzy – szukając sposobów na zrobienie doskonałego dema – przesuwali granice tego, co jest możliwe do uzyskania na amigowym sprzęcie. Jeśli ktokolwiek zna ludzi, którzy byli częścią tej sceny (funkcjonującej do dziś!), proszę o kontakt: [jeremy\\_reimer@hotmail.com](mailto:jeremy_reimer@hotmail.com).\*\*

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

\*Zamboni – potoczna nazwa maszyny do wyrównywania powierzchni lodu, używana m.in. w NHL (przyp. tłum.)

\*\* Jeremy Reimer opublikował swój tekst 13 czerwca 2010 r. Nie wiem, na jakim jest etapie, pisząc swój artykuł. Być może w momencie, kiedy czytacie ten tekst, artykuł już jest skończony. Jeżeli jednak uznacie, że chcecie się z nim skontaktować, pamiętajcie że jego językiem jest angielski :)

# SABRINA 600

## aka A600 Tower MkII

Po pewnym czasie używania mojej „A600 Tower”, opisaney w nr. 5 CAFan, doszedłem do wniosku, że wypada dodać i zmienić parę elementów. Dodatkowo doszły problemy z kondensatorami, które należało wymienić. Oto efekty kolejnych prac nad A600, opisane w tym artykule.

Wspomniane problemy z kondensatorami, miałem już przy pierwszej, uszkodzonej płycie A600, która posłużyła mi na dawcę organów dla gniazda PCMCIA. Po prostu kondensatory C303, C304, wylały elektrolit. Objawy są takie, że podczas pracy Amigi coś zaczyna śmierdzieć, a po rozkręceniu Amigi, widać na ścieżkach i punktach lutowania, naokoło tych kondensatorów, czarny nalot. Do tego C324 i C334, to te przy gniazdach AUDIO, były napuchnięte, więc trzeba także je wymienić. Najlepszą opcją byłaby wymiana wszystkich kondensatorów na płycie głównej, jednak ograniczyłem się do wspomnianych czterech sztuk.



Widok na wylany elektrolit.





Kondensatory po wymianie.

Zastosowałem kondensatory do montażu standardowego, ponieważ w domu nie mam odpowiednich do montażu powierzchniowego, a nie ma sensu specjalnie kupować odpowiedników SMD.

Kolejnym krokiem była wymiana baterii w rozszerzeniu pamięci. Wcześniej zastosowałem jakąś z 386, która po jakichś 3 miesiącach pracy Amigi, przestała działać. Z tego powodu nie miałem RTC w Amidze. Wymieniłem ją na CR2032. W tym celu wlutowałem podstawkę pod taką baterię, dodałem diodę krzemową, żeby nie było problemów przy uruchamianiu Amigi i mogłem cieszyć się ponownym odliczaniem czasu przez Amigę.



Widok na nową baterię.

Pewnej zmianie uległo także wykorzystanie myszki. Po prostu zakupiłem interfejs umożliwiający podłączenie myszki USB. Dzięki temu mogę teraz kupić jakąkolwiek myszkę pod ten interfejs i podłączyć ją do mojej 600-tki. Dodatkowo pojawił się problem z przełącznikiem mysz/joy na porcie 1. Przestał działać prawidłowo, a raczej przełącza się jak chce i kiedy chce. Z tego powodu niemożliwe stało się wykorzystanie interfejsu myszki USB, więc siłą rzeczy zostałem zmuszony do zrezygnowania z tego przełącznika, który i tak nie był wykorzystywany.

Pociągnęło to za sobą konieczność zmiany płytki rozszerzającej, która pierwotnie była wykonana na płycie uniwersalnej. Teraz zastosowałem laminat, na którym dołożyłem kilka dodatkowych rzeczy w porównaniu z pierwotną wersją. Te dodatkowe elementy to:

- interfejs na DF2: i DF3:
- autorfire na joyport 2
- Virus Beeper
- mixer audio CD-Rom/Amiga
- interfejs dla klawiatury od A2000
- gniazdo zasilania AT
- gniazdo RJ-45 z przewodem LAN wewnątrz obudowy (to na przyszłość, jeśli kupię kartę turbo i zwolnię się gniazdo PCMCIA dla karty sieciowej).

Zmieniłem także miejsce podłączenia zewnętrznej klawiatury. Wcześniej miałem druciki przylutowane bezpośrednio do odpowiednich nóżek układu U13. Pod adresem [http://koti.mbnet.fi/udioica/mod\\_a600lang/index.html](http://koti.mbnet.fi/udioica/mod_a600lang/index.html) znalazłem opis przeróbki A600 w obudowę desktop A1200 (bardzo podobna była moja wcześniejsza przeróbka A600, tylko bez CD-Romu). Jest tam opis wykonania interfejsu dla klawiatury od A500. Jednak autor nie lutował się bezpośrednio do scalaka, tylko do odpowiednich pinów na płycie głównej. Rozwiązanie to jest lepsze od mojego, gdyż można lepiej wykonać lutowanie takiego interfejsu. Dlatego zmieniłem ten element w swojej towerce. Dodatkowo jest tam opis, jak zabezpieczyć układ 7427 w klawiaturze A500 przed uszkodzeniem podczas resetowania. Teraz już wiem, dlaczego moja klawiatura A500 przestała działać i musiałem kupić od CDTV.

Po instalacji interfejsu dla 2 dodatkowych flopów, należało domontować dodatkowe sanki dla tychże napędów. Obudowa, którą mam była jakby specjalnie zaprojektowana do tego, żeby obsługiwać 4 stacje dysków. Miejsce dla nich znalazłem poniżej aktualnych, pod paskiem z diodami wskazującymi włączenie komputera i pracę dysku twardego. Należało jedynie dodać sanki poniżej tych, które są w obudowie i wyciąć otwór na dodatkowe napędy.



Widok na dodatkowe sanki w obudowie.

Po tych wszystkich modyfikacjach, musiałem wykonać nową zaślepkę na tylną ściankę.



Przy okazji wewnątrz obudowy okleiłem pianką służącą do wyciszenia komputera. Prawdę powiedziawszy, uczyniłem ten krok w celach czysto estetycznych, ponieważ A600 jest tak cicha, że elementy wydające nieproszone dźwięki to wszystkie napędy podczas dostępu do danych na nośnikach w nich się znajdujących. Nawet wiatrak w zasilaczu nie jest głośny, co tylko poprawia komfort korzystania z tejże Amigi.

Po wykonaniu tych wszystkich usprawnień, mogę teraz cieszyć się swoją sześcietką, która została rozbudowana prawie na maxa. Brakuje jedynie karty Apollo 630, karty sieciowej na PCMCIA, KS 3.1 i interfejsu 4xEIDE. Jednak w tym przypadku zobaczymy, co czas pokaże...

#### Aktualna konfiguracja A600:

|            |  |
|------------|--|
| CPU        | 000/7,14 MHz   |
| RAM        | 2 MB Chip, 2 MB Fast   |
| GFX        | ECS Denise   |
| SFX        | Paula, 4 kanały stereo<br>mikser audio Amiga/<br>CD-Rom  |
| FDD        | 4 x PC Floppy  |
| HDD        | 3,5" - 850 MB  |
| CD-Rom     | Sony x40   |
| Obudowa    | Midi ATX   |
| Zasilanie  | PC AT 200 W  |
| Klawiatura | CDTV, poszukuję A2000/3000   |
| Inne       | Interfejs klawiatury<br>przewód na 4 stacje PC<br>Virus Beeper<br>interfejs IDE 2,5/3,5"<br>kątówka PCMCIA |

Zapraszam do odwiedzenia mojej ciągle rozwijanej strony, opisującej bardziej dokładnie elementy związane z moją A600:  
<http://sabrina600.ca-fan.pl/>

Atreus



nowsze wersje zlokalizowane na nośnikach. Komenda ta nie dodaje plików do archiwum, a jedynie podmienia już istniejące, te ze starszą datą modyfikacji. Komenda bierze pod uwagę ścieżki dostępu, chyba że wyłączymy tą funkcję parametrem (-x)

Lha f archiwum.lha – powoduje „odświeżenie” wszystkich plików w archiwum o nazwie archiwum.lha. Oczywiście porównane są odpowiednie pliki z plikami w bieżącym katalogu.

Lha f archiwum.lha \*.c – wymienia wszystkie pliki w archiwum z końcówkami .c, na pliki o tych samych nazwach z bieżącego katalogu ale z późniejszą datą modyfikacji.

Komenda „h” używana jest w celu porównania i wyświetlenia zmian jakie zaszły w plikach od kiedy zostały dodane do archiwum. Porównuje ona pliki w archiwum z plikami na dysku i wyświetla listę zmian. Opcja (-D) ma trochę inną funkcjonalność dla tej komendy.

(-D0) – domyślnie – każdy zmodyfikowany plik jest listowany po prawej z listą zmian po lewej. Zmiany są oznaczane przez x. Tm – oznacza zmianę daty ostatniej modyfikacji, Sz – oznacza że zmieniony został rozmiar pliku, Pr – zmieniono bity protekcji, Fn – oznacza zmianę komentarza do pliku, Del – plik został usunięty (nie istnieje) z danej lokacji.

(-D1) – każdy różniący się plik jest listowany. Nazwa ukazuje się po lewej stronie a po niej następuje krótki opis co zostało zmienione. Każda dodatkowa zmiana jest ukazywana w osobnej linii.

(-D2) – tak samo jak w (-D1) ale wszystkie zmiany są listowane w jednej linii.

(-D3) – wyświetla jedynie nazwy plików, które się różnią, bez opisów różnic, każda nazwa w osobnej linii.

Jeżeli nie jest podany konkretny katalog lha przyjmie że pliki do porównań znajdują się w bieżącym katalogu.

Lha h ram:uzytki dh0:utilities/ - porównuje wszystkie pliki w archiwum z odpowiednimi plikami w katalogu utilities na dysku dh0: i wyświetla różnice

Lha -x0 h ram:archiwum #?.c dh0: - porówna wszystkie pliki kończące się na .c z archiwum z odpowiednimi plikami w katalogu głównego dysku dh0:

Lha h archiwum.lha - porówna wszystkie pliki z archiwum z odpowiednimi plikami

z bieżącego katalogu

Komenda „l” podaje krótką listę zawartości archiwum. Zawiera nazwy plików bez ścieżek dostępu, rozmiary oryginalny i po skompresowaniu, ostatnią datę modyfikacji i procent kompresji.

Pliki w podkatalogach są oznaczane za pomocą „+” przed nazwą.

Jeżeli chcemy wyświetlić tylko nazwy plików musimy użyć polecenia „lq”. Puste katalogi są wyświetlane jako puste linie.

Komenda „m” zachowuje się tak samo jak komenda „a” z tym wyjątkiem że pliki źródłowe po pomyślnym dodaniu do archiwum są usuwane z katalogów źródłowych.

Lha m archiwum.lha ram:\*.c - przesunie wszystkie pliki z końcówką .c z ram dysku do archiwum.

Komenda „p” działa tak jak komendy „e” i „x” ale komunikaty wysyła do urządzenia stdout. Normalnie są one wysyłane do konsoli lub wskazanego pliku.

Komenda „r” działa podobnie jak komendy służące do aktualizowania lub dodawania plików do archiwum z tym wyjątkiem, że pliki które już są częścią archiwum są zamieniane niezależnie od daty ostatniej modyfikacji pliku. Użycie komendy „u” z parametrem „-Qr” daje takie same rezultaty.

Komenda „t” służy do testowania integralności archiwum. Rozpakowuje ona poszczególne pliki bez zapisywania ich gdziekolwiek. Komenda ta działa wyłącznie na całym archiwum. Nie da się przetestować wybranych kilku plików. Jeżeli zostanie zwrócony komunikat błędu oznacza to że archiwum jest uszkodzone.

Lha t dh0:gry/\* - sprawdzi integralność wszystkich archiwów w katalogu gry na dysku dh0:

Lha t dh0:uzytki.lha - przetestuje integralność archiwum o nazwie uzytki.lha

Lha -R t dh0:\* - sprawdzi integralność wszystkich archiwów na dysku dh0:

Z kolei komenda „u” aktualizuje archiwum. Dodaje pliki których nie było w archiwum i zamienia istniejące starsze pliki. Używana jest tutaj data ostatniej modyfikacji pliku do stwierdzenia który jest nowszy.

Lha u archiwum.lha \*.asm - aktu-

alizuje archiwum o pliki z końcówką .asm z aktualnego katalogu.

Komenda „v” działa podobnie jak komenda „l” ale listuje zawartość archiwum wraz ze ścieżkami dostępu. Komentarze do plików są listowane w osobnej linii i zaczynają się od : (dwukropka).

Komenda „vq” działa podobnie jak komenda „v” ale listuje jedynie pliki które posiadają ścieżkę dostępu. Pomija pliki z głównego.

Komenda „vv” działa podobnie do komendy „v” ale wyświetla wszystkie dostępne informacje w nieco zmienionej formie. Oprócz tego co wyświetla komenda „v” otrzymujemy informację na temat atrybutów, metody kompresji, CRC pliku, DOS ID dla systemu, w którym pliki były kompresowane. Najczęstszymi DOS ID są A,U i M. A dla AmigaDOS, U dla rodziny systemów \*\*IX i M dla MS-DOS. Plik wraz ze ścieżką dostępu jest listowany w osobnej linii. Potem wyświetlany jest komentarz, tak jak przy poleceniu „v”. Można zdefiniować które pliki mają być listowane, w przeciwnym razie wylistowane zostaną wszystkie.

Komenda „x” działa dokładnie tak jak komenda „e”, z tą różnicą, że pliki wypakowywane są zawsze z pełną strukturą katalogów. Czyli tak samo jak „e” z parametrem (-x).

Komenda „y” zmienia parametry wybranych plików w archiwum lub też wszystkich plików w przypadku braku selekcji plików. Czasami komenda ta okazuje się pomocna. Proszę spojrzeć na przykład.

Lha -x0 y dh0:archiwum.lha \*.asm - usunie wszystkie ścieżki dostępu dla wszystkich plików z końcówką .asm w archiwum o nazwie archiwum.lha w katalogu głównym dysku dh0:.

Niestety w tej wersji lha nie da się przepakować archiwów kompresowanych starszą metodą np.: lh1 na najnowszą lh5. Trzeba to zrobić ręcznie. Jednak autor obiecuje dodanie tej możliwości w kolejnych wersjach programu.

I to już wszystkie komendy jakie oferuje ta wersja tego kompresora. Starałem się je opisać jak najprościej dodając przykłady w miejscach gdzie mogły by powstać jakieś wątpliwości. Praktyka czyni mistrza i po dokładnym przećwiczeniu komend, które uznacie za najważniejsze i najpotrzebniejsze nikt nie będzie miał już żadnych problemów z obsługą Lha.

Do następnego odcinka.

MrMat

# Gadżetiarstwo

## i VisualPrefs

### Część pierwsza.

Tuning blatów Workbencha, pod względem wyglądu i funkcjonalności, to jedno z ulubionych zajęć Amigowców. Możliwość pochwalenia się swoim systemowym arcydziełem, w postaci zrzutów ekranu i poddanie go ocenie innym użytkownikom Przyjaciółki, często wynagradza (jeśli opinie są przychylne) trud wniesiony podczas babrania się godzinami w AmigaOS'ie. Jeśli zaś dostaniemy baty, motywujemy się do dalszej pracy. Zresztą na ten temat nie ma się co rozpisywać, wystarczy odwiedzić strony Polskiego Portalu Amigowego, czy Execa'a (a dzięki uruchomieniu „Galerii” w niedalekiej przyszłości także na łamach naszej strony będzie można się pochwalić shotami podrasowanych blatów AmigaOS'u).

System Amigi, jest bardzo elastyczny. Choć sam w swojej podstawowej konfi guracji, porównując go do dzisiejszych wizualnych standardów, jest raczej mało atrakcyjny, to dzięki swojej architekturze dającej możliwość zastosowania dodatków, systemu łątek, programów typu commodity, szybko można zapełnić te estetyczne braki, przeobrażając blat Workbencha w niczym nieustępujące dzisiejszym systemom, cieszące oko środowisko pracy.

No ale jak to się mówi, wszystko ma swoją cenę. I choć na starsze Amigi jest kilka ulepszcaczy (jeden z najsłynniejszych to *MagicWB 1.3*, pozwalający na korzystanie z ośmiokolorowych ikon *MagicWB* w systemie 1.3), to konkretna zabawa zaczyna od systemu 3.0.

Druga sprawa to sprzęt. Niewiele zdziałamy dysponując „gołą” Amigą 1200. Praca kilku łątek, programów typu commodity, czy też innych dodatków, mimo że odbywa się w tle,

pożera cenną pamięć i moc procesora. A przecież nie chodzi tu tylko o odpalenie podrasowanego Workbencha i gapienie się w monitor godzinami, ale o normalną pracę, czy zabawę w ładnie prezentującym się środowisku. Na dodatek, przy słabszym sprzęcie, stosowanie ulepszcaczy może zdestabilizować OS, racząc nas zbyt częstym widokiem GURU.

No dobra, mamy odpowiedni sprzęt oraz odpowiedni system. Do dyspozycji jest masa programów poprawiających wygląd i funkcjonalność systemu Przyjaciółki. Nie będę się tu za bardzo rozpisywał na temat tych aplikacji, gdyż są one znane i stosowane przez Amigowców już wielu lat. Nadmienię tylko, że najbardziej znane to *MUI 3.8*, *MCP*, *VisualPrefs*, *Birdie*, *NewIcons*, *GlowIcons*, kilka łątek na systemowe procedury, np. *FastIPrefs*, *MigicASL*, zamienniki systemowego oprogramowania. A z nowszych np. *PowerIcons*, czy *AmiSTART*.

Wśród wyżej wymienionych znalazł się właśnie także nasz bohater, czyli *VisualPrefs*. Nie chcąc się jednak powtarzać, odnośnie opisywania aplikacji, gdyż jest ona dość znana i popularna wśród użytkowników AmigaOS 3.x, nadmienię tylko że program prócz wielu funkcji bałamucenia procedur systemowych, co przekłada się na dowolną edycję barw w systemie, zmiany wyglądu okien katalogów, czy programów, oraz wielu innych sztuczek, ma możliwość zastąpienia standardowych gadżetów amigowego systemu, plikami graficznymi w postaci małych brushy.

Odpowiednio dobrany pakiet gadżetów, oraz pasująca do niego grafika np. dla *Birdie* (wspomnę tylko, że *Birdie* wrzuca nam wskazany obrazek w obramowanie okien katalogów i programów), daje świetnie wyglądające rezultaty.

I tu, małymi kroczkami dochodzimy do setnej sprawy. Na **Aminiecie**, jest masa gotowych zestawów gadżetów dla *VisualPrefs*, często razem z obrazkami dla *Birdie*. Ja jednak proponuję samemu wykonać taki zestaw, i udowodnię, że nie jest to takie skomplikowane.

#### Na początek kilka słów teorii.

Cała procedura tworzenia gadżetów dla *VisualPrefs* odbywać się będzie w jakimkolwiek programie malarskim. Osobiście polecam *PersonalPaint* (szóstka, lub siódemka). Sam stworzyłem wiele pakietów, przy pomocy tego właśnie programu i na nim będę opierał swój dalszy opis. Do tego może przydać się także jakiś program do obróbki grafiki, dający możliwość precyzyjnego skalowania wycinków; np. *ImageFX*, *ArtPro* czy *Grimm*.

O co chodzi z tym skalowaniem. Bywa czasem, że dany brush nie mieści się w wyznaczonych kryteriach rozmiaru, że brakuje zaledwie kilku pikseli. Dzięki tejże opcji po wczytaniu brusha, możemy wpisać konkretny, interesujący nas rozmiar. Sam *Personal* nie ma takiej możliwości, a precyzyjne skalowanie odbywa się w „locie” (po wczytaniu obrazka musimy go wyciąć i ręcznie skalować za pomocą myszy, co może się okazać mało precyzyjne). W tym przypadku namawiam do skorzystania z *Grimm'a*. Ma małe wymagania sprzętowe, przy tak niewielkich „obrazkach” proces będzie bardzo szybki, no i algorytmy stosowane przy skalowaniu precyzyjnym, nie pozwolą na dużą utratę jakości.

Kolejna sprawa. O ile dysponujemy kartą graficzną, nie musimy się specjalnie martwić o paletę naszych gadżetów. Spokojnie mogą mieć nawet 256 kolorów. Co innego jeśli nasza

Amiga opiera się tylko na kościach AGA. Musimy się zastanowić z jakiej ilości kolorów korzystasz nasz Workbench, ile przypada na menu, ile na ikony, na tapetę, a ile jesteśmy w stanie przeznaczyć na nasze gadżety. Te informacje są Ważne, ponieważ jeśli mamy blat AmigaOS'u np. w 32 kolorach i w całości je przeznaczymy na gadżety, a ich paleta nie będzie się pokrywać z tapetą, wyglądem menu, ikonkami i całą resztą (co notabene jest bardzo prawdopodobne), system Przyjaciółki, podczas uruchamiania będzie ją tak dopasowywać, aby pasowała dla wszystkich elementów. A z doświadczenia wiem, że efekty mogą być, delikatnie mówiąc, dalekie od zamierzonych. Owszem możemy sobie ustawić ekran na maksa (256 kolorów) i nieco „poszaleć” (tworząc gadżety np. w 128 kolorach), ale odbije się to na naszej pamięci graficznej. I w pewnej chwili obudzimy się z ręką w nocniku i sześciuset kilobajtami pamięci CHIP na pokładzie. Przy okazji zaczniemy zawałać fora internetowe pytaniami typu „Przestały mi się uruchamiać niektóre/większość/wszystkie programy. Co robić?”. Po prostu należy znaleźć kompromis pomiędzy wyglądem, a wydajnością. Bazując na własnym doświadczeniu, proponuję ustawić paletę Workbench na 64 kolory. Będzie to optymalne rozwiązanie np. dla Amigi 1200 z 8-ma MB Fastu. W takim przypadku nie powinno być rażących po oczach zgrzytów, pomiędzy ikonkami, tapetą, a gadżetami dla VisualPrefs. No i zostanie trochę pamięci dla uruchamianych aplikacji.

Trzecia sprawa to format i rozmiar naszych plików graficznych, które mają stać się później przyciskami. VisualPrefs dobrze radzi sobie z datatypami, nie mniej jednak proponuję zapisywać brushe w naszym rodzimym formacie. **Jpg**'i lubią się „rozsytać” całkowicie psując naszą pracę, **png**'i długo się wczytują i czasami podczas ładowania gadżetów zawieszają system. Co do **gif**'ów, to szczerze mówiąc nie eksperymentowałem. Od razu wszystko zapisywałem do **iff**'a. Co do rozmiaru, to nie należy przesadzać, ani w jedną ani w drugą stronę. Jeśli brush wyjdzie za mały, to na oknie będzie wyglądał jak rozgnieciona mucha na ekranie monitora. Jeśli będzie za duży, VisualPrefs poobcina go do rozmiarów dopuszczalnych. I tak, rozmiar dla logo wyświetlanego na belce Workbench powinien być nie niższy niż 18 pikseli, oraz nie wyższy niż 26 pikseli, przy czym szerokość jest dowolna (ale radzę nie przesadzać, bo braknie nam miejsca dla komunikatów systemu). Przycisk zmiany ekranów w wysokości powinien odpowiadać obrazkowi logo, a co do szerokości powinien się mieścić między 18 a 26 pikseli. Gadżety na górnej belce okien katalogów i programów: od 18x18 do 26x26 pikseli, przyciski strzałek oraz skalowania okien powinny się zmieścić pomiędzy 8x9

a 30x30 pikseli. Na koniec gadżet znacznika i gadżet cykliczny: dla pierwszego, brush nie powinien przekroczyć 19x19 pikseli, natomiast drugi nie może być większy niż 18x19 pikseli.

I ostatnia sprawa, czyli źródło. Inspiracją dla stworzenia pakietu może być naprawdę bardzo dużo rzeczy. Zrzut ekranu z innego systemu czy programu (z którego możemy wyycinać sobie interesujące nas przyciski), pakiet ładnych ikonek, czy też inna dowolna grafika „nadająca” się do zrobienia gadżetów. A jeśli drzemie w nas, nieodkryta dusza artysty malarza, możemy je sobie sami narysować.

Przystępujemy zatem do pracy. Do poniższego przykładu posłużą ikonki png z pakietu o nazwie *Crystal Clear*. Takie pakiety ikonek możemy znaleźć na stronach poświęconych Linuxowi, a w szczególności środowiskom graficznym np. *KDE*, czy *GNOME* (a ostatnio także na **Aminiecie**). Kolejna sprawa to taka, że często są one posortowane wg. rozmiarów, co ułatwi nam dopasowanie gadżetów, bez czasochłonnego bawienia się w zmianę rozmiarów.

Z racji wyżej wspomnianych utrudnień, dotyczących Amig tylko z kośćmi AGA, skupię się na gadżetach dla 64-kolorowego Workbench.



Wybór brushy, które posłużą jako gadżety powinien być poprzedzony sporządzeniem listy, wg. której będziemy przeszukiwać poszczególne ikonki:

#### a) gadżety systemowe:

*arrowdown* – strzałka w dół  
*arrowleft* – strzałka w lewo  
*arrowright* – strzałka w prawo  
*arrowup* – strzałka w górę  
*close* – przycisk zamykania okna  
*depth* – przełączanie okien  
*sdepth* – przełączanie ekranów  
*size* – skalowanie okna  
*zoom* – zmiana rozmiaru okna

#### b) gadżet cykliczny, znacznika i logo Workbench:

*cycle* – gadżet cykliczny  
*checkmark* – znacznik  
*sbarlogo* – logo Workbench

#### c) gadżety MUI:

*iconify* – minimalizuje program do ikonki  
*MUI* – otwiera preferencje MUI  
*popup* – otwiera menu MUI  
*snapshot* – utrwalenie pozycji i rozmiaru okna.

#### d) inne:

*fallback* – osobiście nie natknąłem się na program wykorzystujący ten przycisk. Zazwyczaj go pomijam i w ogóle go nie tworzę  
*padlock* – myślę że to rodzaj znacznika, pinetki, co do wykorzystywania go jest podobnie jak wyżej.

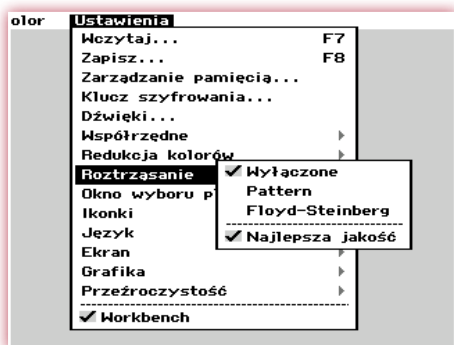
Znając już listę potrzebnych gadżetów przystępujemy do przeglądania ikonek z wybranego pakietu. Tak jak wspomniałem wyżej, posortowanie wg. rozmiarów ułatwi nam wybór, gdyż sugerując się podanymi wcześniej granicami wielkości brushy, nie będziemy zagłębiać się w buszowanie po katalogach z nazwą „128x128”, czy „64x64”. Proponuję od razu wybrać folder „22x22”. Jest to wielkość dość optymalna, spełniająca niemal wszystkie powyższe kryteria. Kolejnym wyznacznikiem, którym powinniśmy się kierować przy wyborze gadżetów, to barwa. Starajmy się je tak selekcjonować, aby miały podobne barwy. Po pierwsze Amiga to nie choinka Bożonarodzeniowa (każdy przycisk w innym kolorze, zdecydowanie będzie sprawiał wrażenie bałaganu i niechlujstwa), po drugie łatwiej będzie nam później połączyć ich palety, jeśli będą do siebie podobne.

Wybrane ikonki przegrywamy do wcześniej przygotowanego folderu. Usuujemy zbędne końcówki (rozszerzenie nazwy mówiące nam o formacie pliku, lub jak w moim przypadku „info”, ponieważ ikonki które wcześniej wybrałem, mimo że pochodzą z pakietów linuxowych i są w formacie png, to wcześniej przygotowano je dla programu PowerIcons, dodając do nazwy rozszerzenie oznaczające amigową ikonkę, zastępując „png”. Dzięki temu, *PersonalPaint* normalnie będzie wczytywać pliki traktując je jak obrazki png).



Po wybraniu ikonki, musimy je tak dopasować do siebie, aby korzystały z tej samej palety, którą następnie będziemy redukować. Pamiętajmy, nasz Workbench ma 64 kolory. Gadżety powinny zmieścić się w 32 kolorach. Dzięki temu, dla reszty pozostanie drugie 32 kolory, no i system będzie mieć rezerwę dla dopasowania całości do kupy. W takim przypadku, efekt powinien być zadowalający.

Wczytujemy Personal'a i ustawiamy ekran na 640x512x8 bitów (256 kolorów). Taka wielkość ekranu, dla kości AGA da wystarczająco dużo miejsca na pracę, a mruganie przy tym trybie nie będzie zbyt uciążliwe. Poza tym sprawdzamy, czy mamy wyłączoną opcję roztrzaskania z użyciem algorytmów „Pattern” i „Floyd-steinberg”. Powinna być zaznaczona tylko opcja „Najlepsza jakość”. Roztrzaskanie wyżej wymienionymi algorytmami nadaje się przy redukcji ilości barw większych grafik, czy całych obrazków. Natomiast przy małych brushach, może spowodować pojawienie się, psującej efekt ziarnistości.



Wczytujemy pierwszego brusha jako obrazek. Ikonki png zrobione są w 24 bitach (16 mln kolorów), choć często zawierają znacznie mniejszą liczbę barw (wystarczy policzyć; 22 razy 22 co da nam wynik 484. I już wiemy, że taki mini obrazek o wymiarach 22x22 piksela nie może mieć więcej niż 484 odcienie. W praktyce zazwyczaj nie przekracza liczby 256 barw). Wybieramy zatem „Format aktualny” i odznaczamy opcję „Przeskaluj”.

Z wczytanego brusha wymazujemy niepotrzebne elementy (prawy klawisz myszy, lub zamalowywanie kolorem tła), zostawiając sam gadżet. Następnie wycinamy go i nadpisujemy na wczytany, ale już w formacie iff. Identycznie postępujemy z pozostałymi. Zdarzyć się może, że podczas usuwania zbędnych fragmentów ikonki może wyjść nieciekawy efekt. W takim przypadku proponuję, albo poszukać innej ikonki „łatwiejszej” w obróbce, lub samemu coś dorysować, dokleić, po prostu pokombinować z wycinkiem.

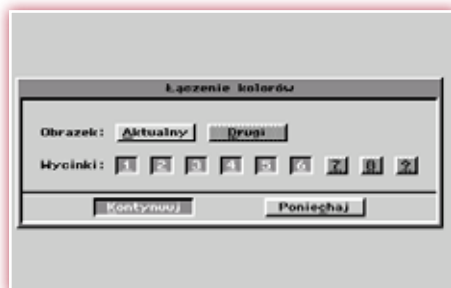
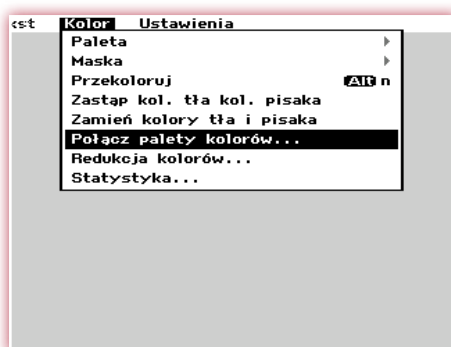
Teraz kolejny etap, nieco trudniejszy. Musimy wszystkie obrazki połączyć tak, aby składały się z tej samej palety barw. W pierwszej kolejności wczytujemy brushe o podobnych barwach.

W Personal'u najeżdżamy kursorem na cyfrę, w lewym górnym rogu paska narzędzi i klikamy prawy klawisz. Otworzy się mały zasobnik, z którego możemy wybrać numer schowka, do którego wczytamy później wycinek.

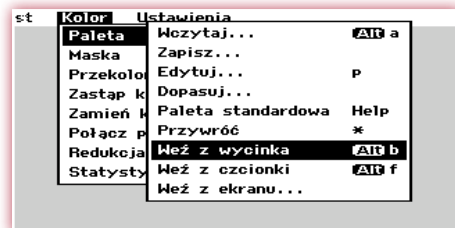


PPaint może przechowywać w pamięci do dziewięciu wycinków. Po wczytaniu gadżetu do nr 1, przełączamy na nr 2 i wczytujemy następnego brusha. I tak postępujemy ze wszystkimi o podobnych barwach.

Następnie łączymy ich barwy, zaznaczając znajdujące się w poszczególnych zasobnikach.



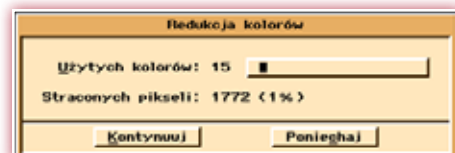
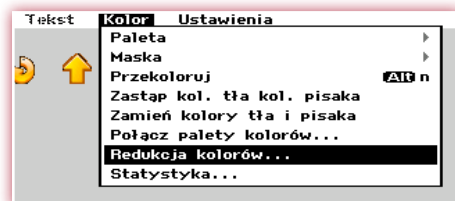
Po tym zabiegu, obojętnie z któregokolwiek brusha, wczytujemy paletę do programu.



Następnie wklejamy je na obszar roboczy. Bywa tak, że po zabiegu łączenia palety w niektórych gadżetach mogą pojawić się jakieś niepotrzebne elementy, np. tło wokół niego. Należy je wymazać.

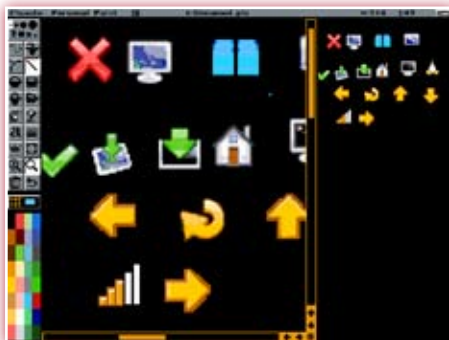
Teraz kolejna rzecz. Musimy zredukować ich liczbę barw, do możliwie jak najmniejszej, przy zachowaniu odpowiadającej nam jakości. Przesuwamy suwak w lewą stronę i obserwujemy podgląd. Gdy uzyskamy efekt, zapisujemy wszystkie gadżety w jednym, ujętym wcześniej, wycinku.

Tą czynność powtarzamy dla wszystkich pozostałych kolorystycznych grup gadżetów.



Wracamy do zasobnika. Wczytamy teraz, podobnie jak wyżej do zasobników, wszystkie wcześniej zapisane grupami wycinki (w moim przykładzie wyszło trzy; szaro – zielona, niebieskawa, oraz grupa gadżetów z dominującą barwą żółtą).

Łączymy, a następnie wczytujemy do PPA-int'a ich wspólną już paletę (identycznie jak w przypadku pojedynczych obrazków, tylko że w tym przypadku do poszczególnych zasobników wczytujemy wycinki z całymi grupami, wcześniej zapisanych przycisków). Wklejamy w, uprzątnięty wcześniej, obszar roboczy i redukujemy liczbę kolorów do 32. Przypatrujemy się i jeśli zajdzie taka potrzeba, ręcznie (pędzlem, kredką, czy gumką) korygujemy wszelkie, powstałe na skutek powyższych zabiegów, psujące efekt błędy w naszych gadżetach.



Zapisujemy wszystko jako obraz, z którego później będziemy już wycinać i tworzyć konkretne przyciski.

Proponuję wyłączyć *Persona* (zwolnić nam to pamięć z niepotrzebnych już, zawalających ją śmieci, nieusunętych z zasobników brushy itp.) i zrobić sobie pięć minut przerwy.

Uruchamiamy *Persona* i wczytujemy nasz obraz z gadżetami. Możemy sobie je posortować i nazwać. Dla potrzeby artykułu poukładałem je w spreparowanej tabelce, ale z doświadczenia wiem, że tworzenie tabelki, jest niepotrzebne i zajmuje dużo cennego miejsca w obszarze roboczym. A przy układaniu gadżetów i tak będziemy musieli ją później wymazać.



Jak zauważyliście na załączonym obrazku, trochę się ich namnożyło. A dlaczego? Już spieszę z wyjaśnieniami. Każdy gadżet musi występować w kilku egzemplarzach. I tak; dla sbarlogo jest tylko jeden obrazek, dla sdepth – dwie sztuki (aktywny i aktywny wciśnięty), checkmark – dwa egzemplarze (aktywny i nieaktywny), oraz cycle – aż cztery (dwa aktywne zwykłe i dwa nieaktywne wciśnięte). Pozostałe gadżety występują w trzech odsłonach (aktywny, aktywny wciśnięty, i nieaktywny). Dobrze by było, aby były różne od siebie. Skąd będziemy wiedzieć, że dany klawisz jest wciśnięty czy nie, lub znajduje się na nieaktywnym oknie programu lub katalogu. Jest kilka szkół mówiących jak to zrobić. Zmiana rozmiaru, pozycji i kształtu (pomniejszenie, obrócenie o kąt 45 stopni, czy przesunięcie o jeden piksel) – metoda nadająca się dla gadżetów wciśniętych,

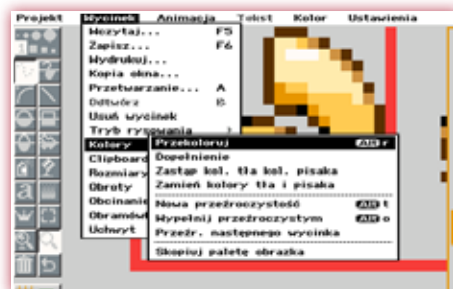
doskonale nadająca się dla pakietów o małej ilości kolorów. Zmiana barw (przyciemnienie, negatyw, konwersja na skalę szarości) – daje dobre efekty dla klawiszy w oknach nieaktywnych, choć wymaga od palety większej ilości barw. W naszym przypadku zastosujemy mix pomniejszenia (dla gadżetów wciśniętych) i przyciemnienia (dla gadżetów na oknach nieaktywnych, w tym celu posłużymy się filtrami *Persona*’a, a kolorów powinno wystarczyć).

Wymazujemy tabelki i robimy czworokątną, dokładnie przylegającą ramkę wokół każdego gadżetu. I w tym momencie pewnie zauważycie, że nie każdy przycisk odpowiada nam, wcześniej ustalonej wielkości (22x22). Niektóre są kilka pikseli mniejsze, niektóre jeden, dwa piksel większe. Jest to wynikiem edycji poszczególnych brushy (usuwanie tła i zbędnych elementów, korygowanie błędów). Wycinamy ramki z przyciskami i wstawiamy, również ściśle, jeden obok drugiej (linie dzielące poszczególne gadżety muszą się nałożyć, tak aby występował między nimi tylko jeden piksel przestrzeni). Kolor ramki dowolny, najlepiej zupełnie odwrotny do tego, z którego składa się dany przycisk.

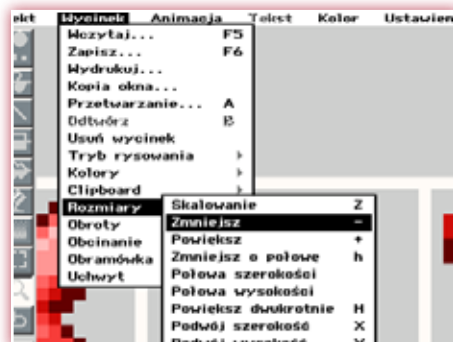


Tutaj nadmienię jeszcze o naszych klawiszach cycle i checkmark. Wychodzi na to, że wyszły za duże. Musimy je pomniejszyć (do wcześniej podanych rozmiarów). Ściśle je wycinamy (każdy osobno) i zapisujemy. Następnie uruchamiamy *Grimma*. Wczytujemy każdy z osobna i zmieniamy rozmiar (w moim przypadku, obydwa przeskalowałem do rozmiaru 18x18). Nadpisujemy je, przechodzimy do *PPaint*’a, wczytujemy, zmieniamy paletę

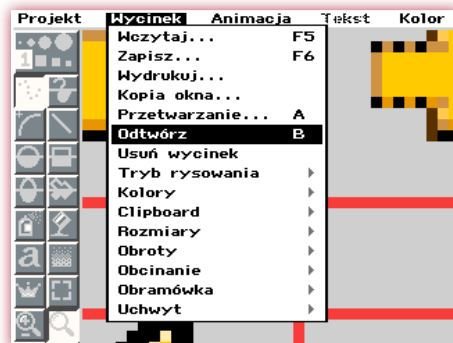
kolorów (podczas nadpisywania w *Grimmie* dochodzi do zmiany kolejności poszczególnych kolorów), wklejamy, korygujemy błędy, wstawiamy wokół ramkę i gotowe.



Teraz pomniejszamy wszystkie wciśnięte (wcinamy je, z menu wycinek wybieramy pomniejsz, lub klawisz minusa). Pomniejszamy je parę razy, aż uzyskamy zadowalający efekt.



Zdarza się, że podczas zmieszania pojawiają się jakieś brzydactwa (niechciane kolory, szpecące efekt piksele). Po wklejeniu pomniejszonego przycisku (oczywiście w sam środek ramki), korygujemy je ręcznie. A jeśli pomniejszyliśmy za dużo, możemy gadżet odtworzyć i zacząć od nowa.



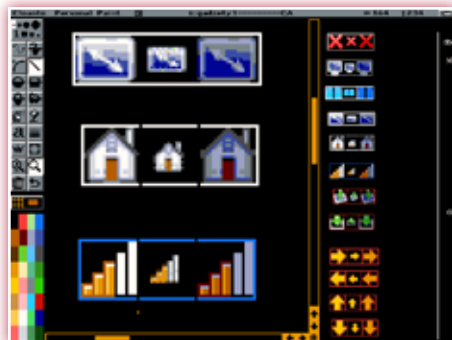
Pamiętajmy, gadżetów sbarlogo i checkmark nie ruszamy. Pora na przyciemnienie nieaktywnych. Z paska narzędzi wybieramy rodzaj filtra (klikając prawym klawiszem my-

szy). Otworzy nam się lista filtrów. Wybieramy Darken -20% i jedziemy (wszystkie nieaktywne, prócz sbarlogo i sdeph).

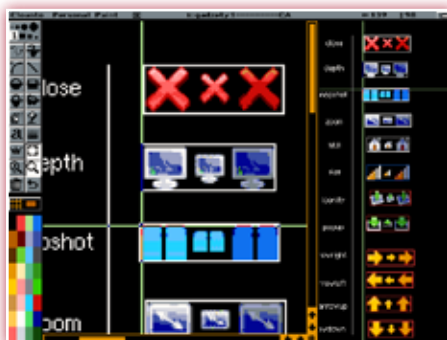


Czas na pasowanie na rycerza. Gdybyśmy je teraz powycinali (nawet pomijając ramkę wokół) VP mógłby trochę zgłupieć i przy wstawianiu, albo je poprzesuwać, albo poprzycinać (w oknie widniałoby tylko pół przycisku). Musimy je w odpowiedni sposób poukładać.

Wymazujemy zatem wewnętrzne linie dzielące gadżety, a następnie wycinamy każdy po kolei i dosuwamy (czy to w lewo, czy prawo) do sąsiadującego. Ważne aby wycinać je wraz z pozostałą ramką. Po tym zabiegu, powstanie tam jeden brush, zawierający w sobie obrazki z gadżetami o równych rozmiarach (wraz z otaczającym tłem). VisualPrefs będzie teraz wiedział, że dany obrazek ma podzielić, w zależności od przycisku, na dwa, trzy lub cztery równe części. I wie także, że każda część odpowiada innemu klawiszowi; aktywnemu, wciśniętemu, czy nieaktywnemu. Czego efektem jest prawidłowe ich wyświetlanie.



Wg. wcześniej wymienionej listy gadżetów, tworzymy puste katalogi o tych samych nazwach. Równy przy ramce (wewnątrz) wycinamy poszczególne brushe i zapisujemy nadając im nazwę „default”. Bez końcówki, czy rozszerzenia. Każdy brush, przedstawiający inny gadżet, zapisujemy do odpowiadającego mu katalogu.



Gdy mamy już zapisane wszystkie gadżety w katalogach, czeka nas ostatni krok do wykonania. Nasze gadżety, nie są równobocznymi czworokątami, tylko nieregularnymi grafikami. Tło zawarte w naszych przyciskach, będzie niestety widoczne w oknach. Żeby tak się nie działo, musimy stworzyć dla naszych gadżetów maski. Dzięki maskom, VisualPrefs, będzie wiedzieć gdzie mają być one wyświetlane (jeżeli kolor tła brał udział w tworzeniu przycisków), a gdzie ma być przezroczyste. Dzięki funkcjonalności Personal'a, zabieg ten jest niezwykle łatwy i przyjemny. Wycinamy wszystkie gadżety (wraz z ramkami, najlepiej cały obrazek) i oczyszczamy ekran roboczy. W menu wybieramy opcję rysowania jednym kolorem oraz kolor nr 1 (następny po kolorze tła) i wstawiamy.



Przyglądamy się rysunkowi i zastanawiamy się gdzie będzie nam potrzebny kolor tła (w moim przypadku jest to czarny). Jeśli w naszych gadżetach znajdą się takie miejsca, zamalowujemy je tym samym kolorem (nr 1). Następnie wycinamy je identycznie jak brushe z przyciskami (ściśle przy ramce, wewnątrz) i zapisujemy, także identycznie jw. (każdy przycisk do odpowiadającego mu katalogowi), nadając im nazwę „default”, ale z rozszerzeniem „mask” (default.mask).

Potem już uruchamiamy preferencje VisualPrefs, wczytujemy katalog z naszymi gadżetami, dostosowujemy szerokości i wysokości na suwakach (aby ładnie wyglądały, niekoniecznie muszą mieć wartości, dokładnie odpowiadające wielkości przycisków) i zapisujemy. Jeśli w naszym systemie buszuje jeszcze Birdie, do naszego pakietu dobieramy dla tej pchełki pasujące patterny i efekt gotowy.



Mam nadzieję, że proces tworzenia gadżetów, z wykorzystaniem ikon, przedstawiłem w dość czytelny sposób. W razie pytań, do Wąskiej dyspozycji jest oczywiście nasze forum, oraz mój adres mailowy.

Serdecznie zapraszam do zapoznania się z drugą częścią artykułu, która zamieszczona będzie w kolejnym numerze C&A Fan. Podpowiem jak samemu narysować gadżety do VP.

Don Rafito

# VirtualCD

Od kiedy Amiga klasyczna „dała” uruchomić się na innej platformie niż sama Przyjaciółka, programiści prześcigają się, aby nasz kochany klasyk uruchamiany na Widnowsie czy Linuxie, był coraz bardziej wierny swojemu pierwowzorowi. Aby oprogramowanie oraz gry, którymi zachwycaliśmy się, będąc w posiadaniu A500 czy A1200, dały się uruchomić na komputerach „jedynie słusznych” (i niestety, jak pokazała historia, górujących na rynku, ale to temat na inny artykuł). Kolejne numerki w wersjach emulatorów, sterowniki do nich i inne dodatki nie tylko podnoszące wierność „udawania” klasyka, ale także wykorzystanie mocy drzemiącej w dzisiejszym hardware, często dają nam możliwość uruchamiania, nie tylko bliskiej oryginałowi (pod względem oprogramowania oczywiście), ale także super szybkiej, daleko zostawiającej w tyle (pod względem sprzętowym) najszybsze motorolki Amigi Classic. I to bez względu czy chcemy použíwać Amigi 500, czy 4000.

Ale jak to się mówi, jeden woli piwo, drugi żyto. Jedni wołają „powolne” 040 i 060, drudzy peceta za 500 zł i WinUae. Ale zarówno, jedni, jak i drudzy mogą korzystać z oprogramowania, jakie powstaje dla klasycznej Amigi, czy to namacalnej, czy emulowanej.

Jednym z takich projektów, który świetnie nadaje się dla emulatorów i klasykowców jest projekt Amikit. Autor projektu, Jan Zahurancik, tworząc go myślał przede wszystkim o osobach emulujących Amigę na pececie. Windowsowy konfigurator, współpracujący z najbardziej popularnym emulatorem WinUae, przygotowuje nam podłoże do zainstalowania mocno wypasionej Amisi na naszym pececie. A po jego instalacji potrzebne nam już tylko będzie systemowe oprogramowanie Przyjaciółki. I możemy cieszyć się już high-endowym klasykiem, z gotowym systemem pełnym świetnego softu.

I to na tym sofcie, chciałbym skupić swój cykl. Autor Amikit'a, prócz udogodnień, wspomnianych wyżej, zawarł w pakiecie spory zbiór programów użytkowych, z których mogą skorzystać nie tylko entuzjaści Emu-Ami, ale także osoby posiadające prawdziwego klasyka. Można tam znaleźć edytory tekstu, programy graficzne i muzyczne. Natrafimy także na przeglądarki grafiki, stron internetowych, czy dokumentów w formacie pdf.

Większość tych aplikacji możemy znaleźć w sieci (m. in. na Aminecie), ale sięgając po ten

pakiet, bez konieczności szukania, wertowania wielu stron, odwiedzania różnego rodzaju archiwów poświęconym oprogramowaniu dla Amig klasycznych, praktycznie od razu mamy do dyspozycji zbiór programów, który przeobrazi naszą Przyjaciółkę w narzędzie przydatne nie tylko do odpalania kultowych gier.



Program, który chcę zaprezentować w pierwszej części cyklu, na pewno przyda się osobom magazynującym na swoich krążkach, twardych dyskach, czy jeszcze innego rodzaju nośnikach obrazy płyt CD i DVD, w formacie ISO. Czyli, można powiedzieć, niemalże wszystkim. Możliwość oszczędzania miejsca, jaką daje format ISO to jedna z wielu jego zalet. Dzięki temu formatowi, na jednej płycie DVD można pomieścić nawet sześć krążków CD. A jeśli mamy do dyspozycji większy nośnik, także ISO z płyt DVD.

No dobra, a co jeśli chcemy skorzystać z takiego obrazu płyty, aby coś z niego skopiować, czy przejrzeć zawartość. Jest kilka sposobów. Można np. bawić się w pisanie mountlisty, podczepić pod system dosdriver, szukać w sieci jakichś filesystemów do obsługi tego standardu. Można po prostu taki obraz wypalić na krążek, ale jaki byłby sens w tym całym magazynowaniu. Najlepiej skorzystać z prostego, przejrzystego i banalnego w obsłudze programu, który podczepi nam dany obraz płyty CD, lub DVD, jako dodatkowy napęd w AmigaOS. Takim programem jest właśnie „VirtualCD”.

Tak jak napisałem, prosty, banalny i przejrzysty. Nie musimy nic grzebać w systemie, nie musimy pisać mountlist, ani instalować nowych filesystemów. Wystarczy Amiga na której chodzi minimum Amiga OS 3.5, oraz CacheCDFi-leSystem (który od 3.5 znajduje się już w OS'ie). Wszystkie „bebechy” potrzebne do zadziałania programu zawarte są w jego katalogu.

Obsługa aplikacji, łatwizna. Uruchamiamy „VirtualCD”, klikamy na gadżet przy iso file. Otworzy się nam okno wyboru. Szukamy naszego pliku obrazu. Po jego wyborze klikamy na gadżet Mount Device. W programie pojawi się nam informacja, że plik został zamontowany, a na blacie Workbenchu pojawi się nam ikonka nowego napędu. Ot i cała filozofia.

Zmiana płytki, nie ma sprawy. Klikamy na Dismount Device, a reszta tak jak wyżej. Jeśli nie pozostał w pamięci jakiś task uruchamianego programu z podmontowanej płytki, ikona napędu znika.

Całą operację bawienia się z dosdriverami, mountlistami przejmuje program. Podmontowany plik ISO widziany jest w systemie jako napęd o nazwie „VCD0”. I choć w nazwie programu widnieje „CD”, to program obsługuje także obrazy płyt DVD.

Dzięki temu, że „VirtualCD” korzysta z OS 3.5/9, mamy możliwość zikonifikowania programu, aby nie zajmował miejsca na blacie, podczas gdy my będziemy buszować po naszych pomagazynowanych cedekach.

Sam program niewiele waży i nie zajmuje sporo zasobów systemowych, gdzie przy niektórych programach tego typu (znanych z innych platform) jest to niemożliwe, gdyż podmontowane ISO, w całości ładowane jest do pamięci komputera. Autor podaje, że program do uruchomienia wymaga 2 MB. A podczas buszowania po katalogach podmontowanych krążków zauważyłem zużycie pamięci sięgające 3-4 MB.

Prosty w obsłudze, nieskomplikowany, przejrzysty i szybki. Do tego AmigaOS 3.5 (gdzie spora część Klasykowców przesiada się lub już przesiadła się na 3.9) i niewielkie wymagania. Tyle za program w pełni darmowy, dający możliwość korzystania ze zmagazynowanych krążków w formacie ISO. Oszczędność miejsca i czasu. Nic dodać, nic ująć. Ten program trzeba mieć.

**Don Rafito**

**Gatunek:** device/cdrom  
**Producent:** Michael Lanser  
 (www.lanser-online.de.vu)  
**Dystrybucja:** Aminet, Amikit  
**Rok:** 2002  
**Ilość dysków:** 75 KB, HD  
**Platforma:** 020, AmigaOS 3.5, CachesCDFs

**Ocena: 100%**

# WYWIAD Z LIFTER'em

Poznajcie historię człowieka, który był członkiem zespołu tworzącego jeden z najlepszych magazynów dyskowych na Amigę, w latach świetności tego komputera. Jego karierę od Spectrumba do Amigi, oraz kulisy powstawania wspomnianego magazynu dyskowego. Postać znana w świecie Amigi jako Lifter, a w czasach działania na C64 znana jako Pumex. Zapraszamy do wywiadu.

*Dla tych co nie wiedzą kim jesteś może na początek powiedź trochę o sobie.*

Najpierw grałem. Potem zacząłem troszkę „dłować” programy na 8-bitowce. A potem odkryłem Amigę i scenę na nią... i się zaczęło DZIAĆ. Pamiętacie zin *Fat Agnus? I Liftera/True Genius?* Bo jeśli tak, to witam starych znajomych.

*Tajemniczy jesteś :) Może zacznijmy od samego początku, czyli od pierwszego komputera. Jak się zaczęła twoja przygoda z komputerami?*

Pierwszy komputer, jaki widziałem to ZX-81. 1 kB pamięci, a jakże. Byłem pod wrażeniem. Ale minęło kilka lat zanim dorobiłem się używanego ZX-Spectrum 48 K. Do dziś pamiętam, że kosztował mnie jakieś 6 miesięcznych pensji. W ogóle tak się potem składało, że co kupowałem kolejny komputer, to zwykle wydawałem półroczne dochody (sprawdzało się to aż do pierwszego PC). To był początek 1986 roku. Jaka radość! Nie starczyło już na magnetofon, więc używałem prastarego Kasprzaka, który w dodatku miał wyjście cinch, ale nie do końca zgodne ze standardem. Krótko mówiąc, kabelek w nim „siedział” luźno, nie kontaktował. Do tego magnetofon lubił wkręcać taśmę. Wczytywanie wyglądało więc mniej więcej tak, że trzymałem magnetofon na kolanach, jedną ręką dociskałem kabelek do gniazda, a drugą trzymałem na wyłączniku, żeby - gdy zobaczy-

łem że jedna rolka taśmy się nie kręci, co było sygnałem, że magnetofon wkręca taśmę - go wyłączyć. To, że za trzecim-czwartym razem gierka się wczytywała... to cud. Tak się składało, że jak dostałem komputer, to panowała epidemia grypy. Polecałem więc do lekarza, pokaslałem, pochrypiałem... i dostałem tydzień zwolnienia. Spędziłem go na graniu. Spałem przez ten czas może przez 10 godzin w sumie. Po tygodniu poszedłem do lekarza - ten spojrzał na moje przekrwione i podkrążone oczy, bładą i napuchłą twarz, zmierzwił włosy i stwierdził z lekką wystraszony, że chyba mam jakieś komplikacje i dołożył jeszcze pięć dni. Jak myślicie, jak spędziłem ten czas? No, ale tu już udawało mi się spać po 4-5 godzin dziennie. A i tak nie ukończyłem cholernego Knight Lore. Do dziś mnie szlag trafia, jak sobie to przypomnę. Potem kupiłem sobie nowy magnetofon, ze standardowym cinchem, nie wciągający taśmy... Byłem królem.

Do dziś pamiętam cotygodniowe wyjścia na giełdę komputerową na Wyspiańskiego. Początkowo mieściła się w takim małym budyneczku, gdzie dosłownie łaziliśmy sobie po głowie. Jedna kaseta z gramami do Specy kosztowała 1000 zł. Dodam, że zarabiałem wówczas 13 500 zł miesięcznie (wcale bym się nie obraził, gdybym znów tyle zarabiał). Co tydzień kupowałem sobie jedną kasę, więc szło na to 1/3 pensji. To była pasja. Tamże kupiłem joystick Quick-shot. Chyba za 2 500 zł. Te ceny...

## FAT AGNUS VIII



THE TRADER/INU

Sprzedawca - pamiętam - powiedział, że jak będę go szanował, to może popracować NAWET rok. Mam go do dzisiaj. Nadal działa, choć już ledwo-ledwo. Zresztą stoi na biurku, jako przycisk do papieru i robi za pamiętkę.

Pewnego razu dostałem premię - 2 000 zł. Od razu polecałem na giełdę. Ach, dwie kasety, ale będzie grania! Biorę dwie ze stolika - a kupowałem u niejakiego Plomby, który robił wtedy za giełdowego rekina. Miał taki szpaaaanerski dwukasetowiec, na którym kopiował kasety (albo pojedyncze programy) z kasety-matki na drugą, na przyspieszonych obrotach, dzięki czemu mógł ich trzaskać całe mnóstwo. No więc biorę te dwie kasety, a on: 2 400 zł. JAK TO @#\$\$%^ 2 400?! „Podrożało” - wrzucił ramionami Plomba. Wku... na maksa przeleciałem się po całej giełdzie, ale wszyscy solidarnie trzymali te same ceny. Normalna zmowa cenowa. To kupiłem sobie jedną i wracałem z giełdy wściekły okrutnie, bo jak się człowiek nastawił na 2 kasety (co dawało 20 gier), a kupił jedną, to czuł się wręcz okradziony. „Cholerni handlarze, wyzyskiwacze...” - mruczałem. I nagle mnie olśniło. Cholera, toż zamiast WYDAWAĆ kasę, to ja mogę na tym ZARABIAĆ! Toż miałem już w domu z 20 kaset, więc z 200 gier, a to już niezły punkt wyjścia na jakiś niewielki biznes! I to była chwila, która zmieniła moje całe życie. Serio.

*Jak wyglądało takie handlowanie na giełdzie? Były z tego większe pieniądze?*

Ja może zacznę od tego, że choć udzielałem się na giełdzie, to tak naprawdę był to tylko dodatek do innego wykorzystania komputera - scenowania, grania, pisanie i tworzenia muzyki (albo raczej wyrobów muzyko-podobnych :)). To z tego powodu zmieniałem sprzęt, a nie dlatego, że chciałem więcej zarobić...

Ale wróćmy do giełdy. Jak już wiecie - postanowiłem zarabiać na giełdzie. Dobrałem sobie współnika (szybko odpadł), umówiliśmy się, że każdy zrobi po 10 kaset i pojechałem na giełdę, ale nie niedzielą, a sobotnią, zorganizowaną w jakiejś szkole. Mój współnik zrobił całe 3 kasety, ja 10, rozłożyliśmy stoliczek i zacząłem karierę. Plomba, pamiętam, miał dość średnią minę, gdy jeden z jego najlepszych klientów zaczął mu robić konkurencję... ale zniósł to po męsku. Zresztą „z czym do gościa” - on był rekinem, my płoteczkami. Debiut wypadł średnio, bo nic nie sprzedaliśmy (inna rzecz, że ruch był malutki), a do tego kupiłem od Plomby nowe gierki (siła nawyku). Ale na drugi dzień pojawiliśmy się na Wyspiańskie-go... i zarobiłem pierwszą kasę. Ha! Jaki byłem z siebie dumny.

Teraz mała dygresja. Wiecie jak wyglądało tworzenie kasety z gram na ZX Spectrum w sposób „klasyczny”? Mordęga! Wgrywał się bodaj Copy-Copy), potem grę z kasety, następnie zgrywało się to na inną kasetę. Do tego niektóre gry były duże, a kopier zajmował miejsce w pamięci, więc... robiło się to na dwie raty! Jedną grę na dwie raty. Stworzenie jednej kasety zajmowało więc co najmniej 2 godziny. Etykiety robiłem na maszynie do pisania, a do tego malowałem mazakami - ręcznie - taką spectrumową skośną tęczę, żeby ładniej wyglądało.

Dodam, że ogromnym problemem były kasety, których... nie było. Pamiętajcie, że to był czas socjalizmu i w sklepach jedyne, co było w dużej ilości, to klienci. Po kasety jeździłem do Gorzowa, bo tam „Stilon” (jeden z 2 producentów kaset w PRL) miał sklep firmowy. Przy czym i tam nie było lekko, bo w sklepie obowiązywała zasada, że „odbiorcy indywidualnemu” sprzedawano 10-15 kaset „na głowę” góra. A ja potrzebowałem z 50, a potem dużo więcej, bo wyprawa do Gorzowa zajmowała dzień czasu, więc jeździłem tam nie za często. Nie będę opisywał, jakie cuda wyczyniałem, żeby wydębic „hurtową” ilość kaset, ale powiem, że się udawało za każdym razem. Zawsze miałem „gadane”.

Ale ad rem. Poświęcając dziennie 2-4 godziny mogłem stworzyć 1-2 kasety, czyli tygodniowo tworzyłem ich z 10-12. Schodziły „na pniu”, a dodając do tego to, co zarabiałem kopiując programy na giełdzie, to wychodziło mi, że w jedna niedzielę zarabiałem mniej więcej tyle, co w pracy w miesiąc (dodam, że zarabiałem kiepsko, sporo poniżej średniej krajowej).

Można więc powiedzieć, że miesięczny dochód z giełdy to była mniej więcej równowartość 2 średnich krajowych. Dużo czy mało? Przypomnę, że wtedy jak człowiek zarabiał z 30-40 \$ (w przeliczeniu) to uważał, że zarabiał nieźle... Więc ja ze 100 \$ miesięcznie byłem paniskiem.

W każdym razie po 2-3 miesiącach stać mnie było na magnetofon dwu kieszeniowy za całe 100 \$ (kupowany w Peweksie - starsi wiedzą, o co chodzi, młodszym powiem, że to był sklep, gdzie można było bez kolejki kupić bardzo fajne rzeczy - ale za dolary, których jednak w PRL nie wolno było kupować :)). To znacząco rozwinęło mój biznes, bo teraz skopiowanie kasety wyglądało tak, że w jedną kieszeń wsadzało się „mastera”, w drugą czystą kasetę i włączało się kopiowanie na przyspieszonych obrotach. W efekcie jedna kasetka wychodziła w 30 minut, a nie 2 godziny. Tak samo można było kopiować pojedyncze programy - już bez użycia komputera.

Ale pechem jest, że kiedyś na giełdzie usłyszałem jak gra Commodore 64 (tzn. jego możliwości muzyczne). No i umarłem. Bo ja zawsze chciałem być muzykiem. Tzn. Bozia dała chęci, ale nie dała talentu. A możliwości muzyczne Spectruma odpowiadały pecetowskiemu PC-Speakerowi (choć ZDOLNI muzycy potrafili wyczarować cuda z niego). Najpierw kupiłem sobie taką specjalną przystawkę do Spectruma (układ AY), który dawał możliwości brzmieniowe zbliżone do Commodore. Byłem chyba w pierwszej piątce Polaków, która toto miała, bo była strrrasznie droga (no, ale na pasję kasy nie żał...).

Ból w tym, że istniał do tego urządzenie jeden program muzyczny, do tego piekielnie skomplikowany i nie potrafiłem go rozgryźć. W końcu się wkurzyłem, sprzedałem Specę i to, co miałem do niego, dołożyłem to, co wcześniej zarobiłem - i starczyło akurat na C-64 ze stacją dysków. Byłem przekonany, że jak będę miał komputer z dobrym układem dźwiękowym i przyzwoitym softem, to zacznę tworzyć lepszą muzykę. Naiwniak ze mnie. Tu pozwolę sobie na małą dygresję: czasem jak oglądałem program z utalentowanymi ludźmi, gdy się ich pyta o to, w jaki sposób np. potrafią tak znakomicie grać na skrzypcach, pada odpowiedź „10% talentu i 90% ciężkiej pracy”. To zapewne prawda, ale żaden nie powie tego, co ja Wam teraz: gdy nie masz tych 10% talentu, to ciężką pracę możesz sobie w d... wsadzić. Bo ja intensywnie pracując nad opanowaniem trackera na C-64 i próbując komponować, tworzyłem tylko coś, co w najlepszym wypadku nadawało się do odstraszania kretów na działkach czy płoszenia ptaków na lotniskach. Ale twardo wierzyłem, że kiedyś się nauczę tworzyć fajną muzykę. A potem usłyszałem Amigę... i byłem pewien, że jak ją kupię, to wtedy

już na pewno... ale do tego jeszcze wrócimy. W każdym razie, nie mogąc tworzyć fajnej muzyki, grałem, handlowałem (dorobiłem się drugiej stacji dysków) itd. W tym czasie spotkałem też pierwszych przedstawicieli sceny komputerowej. Do tego też wrócimy. W każdym razie na C-64 na wrocławskiej giełdzie liczyło się trzech ludzi: Mirek Zapędzki, Waldi Czajkowski i ja. W tej kolejności. A dokładnie: Mirek, Waldi - duża przerwa - ja ----- duuuuża przerwa --- i reszta. Chociaż byliśmy konkurentami, panowała między nami bardzo fajna koleżeńska atmosfera. Jeśli to czytają, to obu serdecznie pozdrawiam! Swoją drogą o tym, że Mirek jest ojcem olimpijczyka dowiedziałem się dopiero teraz, z twojego wywiadu! Dodam jednak, że Mirek i Waldi byli „handlowcami” pełną gębą, mieli wielkie stoiska i praktycznie z tego żyli. Ja miałem dwa stoliki i traktowałem to jako źródło niezłej kasy, ale nie wiązałem z tym znaczących życiowych planów. Pracowałem gdzie indziej, a handel to był tylko dodatek do mego życia. Dodam, że na wrocławskiej giełdzie spotkałem (też przy stoliczku, jako sprzedającego) pewnego Bardzo Znanego Polskiego Programistę, autora gry na PC, która zrobiła furorę na Zachodzie. Nie podam nazwiska, bo może nie chce teraz takiej reklamy. A jak pojechałem do Wawy po nowe gierki, to na Grzybowskiej widziałem też handlującego osobnika, którego po wielu latach spotkałem - ja wtedy byłem redaktorem CDA, a on szefem Dużej Firmy Dystrybuującej Gry (również pominę nazwisko). Popatrzyliśmy na siebie z lekkim i przyjemnym zdziwieniem. A jak chcieliśmy zrobić sobie reklamę w TV, to szefem firmy tworzącej owe reklamy był... jeszcze inny znajomy z giełdy. Człowiek, który chyba pierwszy w Polsce (a na pewno we Wrocławu) miał Amigę. Mafia, panie, mafia...

Ale wróćmy do handlu. Jako że Mirek i Waldi przewyższali mnie potencjałem, rozkręciłem błyskotliwe działania marketingowe, by to zneutralizować. Waldi miał wielki transparent: „5000 gier na Commadore”. Ja wywiesiłem: „Tylko 500 gier, ale za to same nowości”. Waldek przyszedł potem z ciut kwaśną miną i zaczął narzekać, że się z niego klienci śmieją, że ma same starocie. To ja wymyśliłem katalogi z opisem gier (żeby nie musieć każdemu klientowi osobno tłumaczyć, o czym jest dana gra) i parę innych zagrań też, włącznie ze zrzućaniem gier dyskowych (a raczej ich części) na kasety, żeby posiadacze magnetofonów mogli sobie w nie pograć (opatrzone zresztą dumną informacją „cracked by...” :)). Kasa płynęła niezłe, nie powiem. Pensja z pracy była moim kieszonkowym. Ale bywały też przygody - raz np. praktycznie całą giełdę zwinęła milicja. Nie, nie za handel „piratami”. W PRL to było le-gal-ne... Szło o niepłacenie podatków od tego. Ale byliśmy zdziwieni - to są jakieś podatki?! Tu dodam, że nie dałem się zwerbować

jako „informator” panom z MO w zamian za „odpuszczenie grzechów”, więc przekazali moją sprawę do Urzędu Skarbowego i miałem „trochę” problemów. Na szczęście jakoś udało mi się wyłgać - ale musiałem się zarejestrować i płacić owe podatki. Dodam, że teoretycznie, aby sprzedawać programy na giełdzie to powinienem wówczas mieć „wykształcenie kierunkowe (informatyk!) i 3 lata praktyki w handlu”. I jak ten system mógł nie upaść? Miesiąc czasu zajęło mi przekonanie urzędników, że aby skopiować program i wydać resztę, nie muszę być Einsteinem. Z innych ciekawych wspomnień - „zatrudniłem” do pracy na giełdzie pewną panią, z którą w swoim czasie wiązałem spore nadzieje i plany. I choć początkowo przyniosło mi to zysk (bo na 4 ręce i dwie buzie szło dużo szybciej, a że była ładna, to przyciągała chłopaków do mego stoiska), to finalnie byłem w plecy, gdyż po jakimś czasie okazało się, że osobnik siedzący niedaleko (wtedy już były PC) wydał się jej bardziej interesujący niż ja. Cóż, tak to bywa. Dodam, że do dziś z tą panią utrzymuję bardzo dobry, acz tylko przyjacielski, kontakt. Ale przez czas jakiś PC-tów i ich posiadaczy jakoś tak nie lubiłem, nie wiedzieć czemu. Z drugiej strony - dzięki temu wydarzeniu mogłem spotkać inną panią, która od wielu lat jest moją żoną... W międzyczasie zaangażowałem się troszkę w tworzenie maga na C-64 - niestety dziś już nawet nie pamiętam jak się nazywał. Tzn. pisałem do niego teksty. Spodobało mi się to. A niejaki „Pampam” i spółka zrobili mi fajne interko do „crackowanych” przeze mnie gier. Za 100 000 zł, bandyci. Nie wstyd wam było tak zdzierać? Do dziś nie zapomnę miny pewnego kodera, który powiedział mi: „zrobiłem fajne intro”, a ja naiwnie się zapytałem „pod czym” (tzn. pod jakim progra-

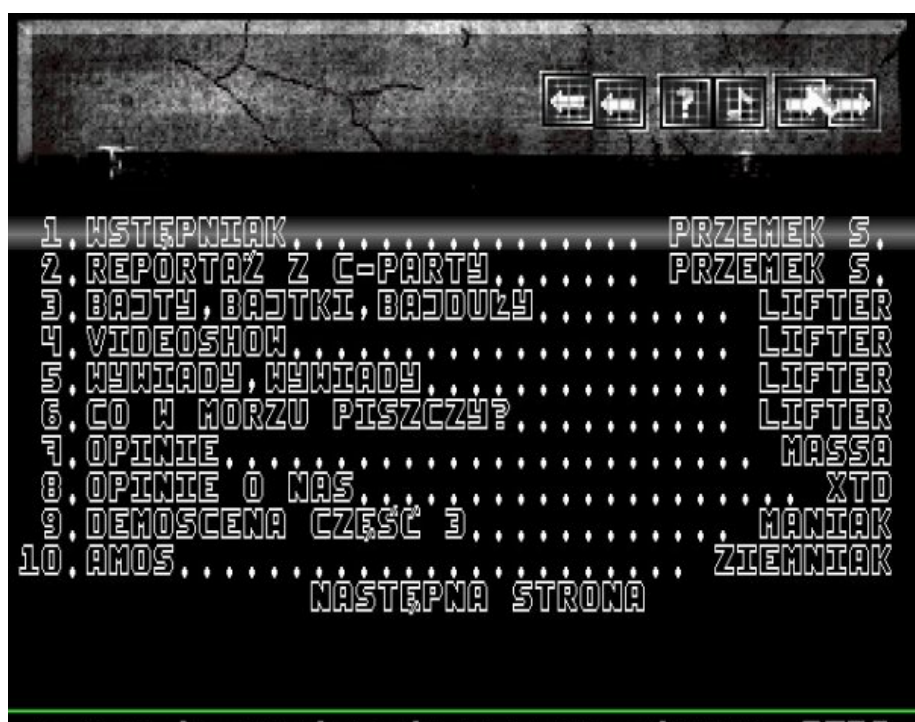
mem, bo były takie, do tworzenia interek - skąd mogłem wiedzieć, że on je tworzy od podstaw, w assemblerze). Gdyby spojrzenie mogło zabić... W każdym razie wtedy się dowiedziałem, że są ludzie, którzy programują, tworzą muzykę itd. i robią to po prostu dla przyjemności na tzw. „scenie komputerowej”... I tak parę lat przeminęło, nastał kapitalizm. Doszedłem do wniosku, że trzeba „wziąć sprawy w swoje ręce”. Zwolniłem się z państwowej pracy, dołożyłem to, co zarobiłem na giełdzie i założyłem firmę handlującą sprzętem komputerowym, nazwaną „Interface”. Z rozpędu jeszcze handlowałem z pół roku programami na giełdzie, ale to już był schyłek mej nielegalnej działalności, gdyż czasu nie starczało, obroty spadały, a i w końcu

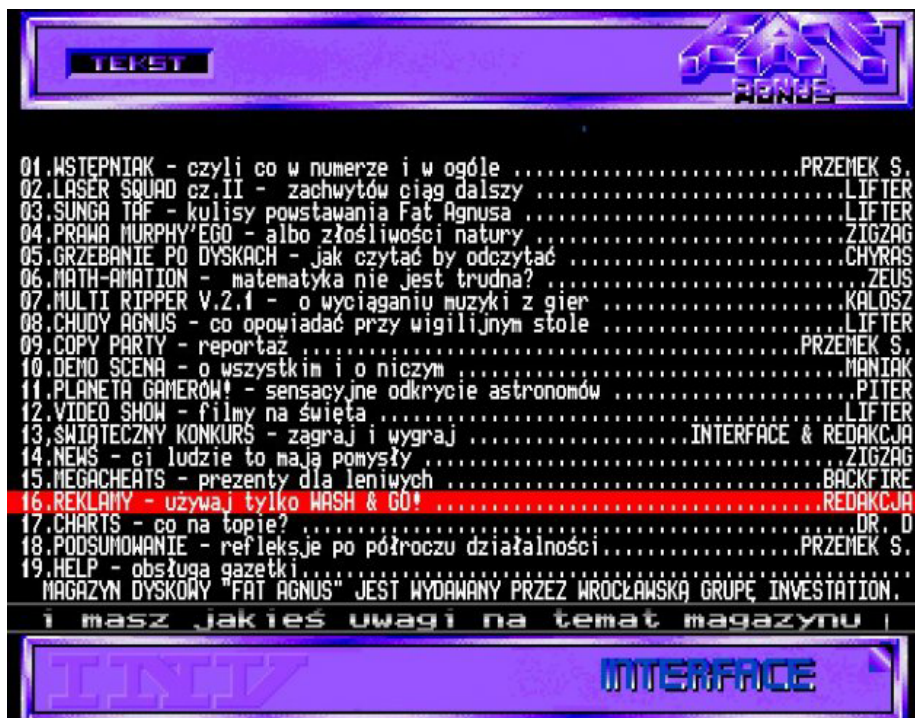
wyszła ustawa o prawie autorskim i sprzedaż cudzego softu stała się nielegalna. Tak i wtedy ściągnąłem piracką flagę z masztu i postanowiłem: „nigdy więcej”. I słowa dotrzymałem - od tej pory nie tylko nie sprzedawałem, ale i nie kupowałem pirackiego softu. A giełda na Wyspiańskiego znów stała się tylko i wyłącznie nudną stolówką dla studentów Politechniki. Jakiś czas temu tam zajrzałem, stanąłem w miejscu, gdzie miałem kiedyś swe stoisko. Ale w powietrzu był już tylko zapach pierogów ruskich i kompotu, klimat giełdy odszedł i nie wróci. Swoją drogą, ciekawe co robi Plomba? Mirek miał kiedyś sklep komputerowy, potem, zdaje się, hurtownię z farbami, a później już kontakt się nam urwał. Co robi Waldi - już wiem. Przy okazji pozdrawiam wszystkich, którzy u mnie kupowali sofci albo chociaż pamiętają interka z „Pumex” na początku gry czy nagłówek „cracked by Lifter”.

#### A jak zainteresowałeś się sceną komputerową i jakie były twoje na niej początki?

Kurczę, teraz już ciężko powiedzieć „jak”. Zapewne, będąc na giełdzie komputerowej, natknąłem się na koleśki oglądających demo i się zainteresowałem „co to jest?”. Tak myślę. A początki... Te na C-64 pominę, bo ograniczały się do okazjonalnej współpracy z twórcą Micrus-Copy - tzn. pisałem teksty do jego maga (nazwa wypadła mi już z pamięci) i robiłem intra do gier na demo-makerze.

Potem nadeszła Amiga. Ludzie zaczęli się organizować w grupy, więc też chciałem, ale o programowaniu nie miałem pojęcia, grafiką nawet nie próbowałem się zająć, bo wiedziałem, że jestem antytalentem. Przez długi czas wydawało mi się, że jestem muzykiem, ale cóż - jak już mówiłem, Bozia dała chęć, ale





nie dała talentu. Oglądając demo jakieś tam czytałem informacje w lejącym u dołu scrollu. Było widać, że ten, co tam pisał nie tylko nie wiedział, co pisać, ale i pisać nie potrafił (pewnie programista :P). I tu mnie olśniło - mogę pisać im te teksty! Ale okazało się, że świeżutka wrocławska grupa Investation postanowiła wydawać magazyn dyskowy. Nawet im wymyśliłem nazwę „Fat Agnus” (od nazwy chipa w Amidze). No to sprawa była jasna. Zacząłem pisać teksty, potem robiłem też korektę, a po jakimś czasie zostałem edytorem tego maga - który potem stał się największy w Polsce i jednym z większych w sieci, a przeżył dobrych 5 lat, mając coś koło 20 edycji.

Pochwalę się też, że zająłem na Copy Party w Warszawie, w bodaj 1995 r., III miejsce w music-compo, ku swemu wielkiemu zdziwieniu zresztą. Głównie dlatego, że chłopaki przygotowali jakieś ogromne monumentalne utwory, symfoniczno-heavymetalowo-jazzowe, a ja dałem 2-minutowy żarcik muzyczny „Kto wrobił królika Arnolda” (czy ktoś ma ten moduł?) i poprzez kontrast z kolejnym 7-minutowym gniot... poematem wydał się ludziom inny i został dostrzeżony. Na tymże compo Scorpik zajął II miejsce, więc potem chodziłem za nim, klepałem go po ramieniu i mówiłem „my, muzycy...” czy „wiesz, jak niewiele nas dzie- li...” itd. Scorpik zgrzytał zębami, ale cierpiał w milczeniu. Po tym wydarzeniu zakończyłem swą karierę jako muzyk (żeby nie psuć sobie statystyki :P), a jako edytor FA przetrwałem do 1996 roku, gdy z tworzenia zina awansowałem na twórcę komputerowego pisma. No, jeszcze w międzyczasie odrobinkę swappowałem, ale to był friendly-swap, tylko raz mnie ścignęli za fakowanie stampów, a i to niegroźnie.

I to zasadniczo cała moja scenowa kariera...

**Jak wyglądało redagowanie i tworzenie**

**magazynu „Fat Agnus”?**

Wiesz, z tym redagowaniem FA, to sprawa prosta i nieprosta. Teoretycznie sprawa jest prosta - ludzie nadsyłali tekst, ja to obrabiałem, dawałem koderowi, muzyk dorzucał muzykę, grafik okładkę... i już. W praktyce stale ści-gałem ludzi o teksty, koder cały czas grzebał w kodzie magazynu i w efekcie było tam tyle poprawek i łatek, że po jakimś czasie sam się przestał w tym orientować, więc za diabła nie potrafił pojąć, dlaczego mag nie chce działać, choć powinien, a potem czemu działa, choć NIE powinien. Muzyk zalegał z muzyką, grafikowi się nie chciało zrobić grafiki... Ale potem nagle wszystko było i kolejny numer szedł do ludzi.

Jako anegdotkę dodam, że ekipa w swoim czasie zbierała się na zapleczu mojej firmy (miałem wówczas sklepik ze sprzętem komputerowym) i tam teoretycznie składała numer, a w praktyce grała w Laser Squad na C-64 albo po prostu pitoliła. Gdy już wszystkie prośby zawodziły, miałem sposób: „albo się bierzecie do roboty, albo puszcze wam parę swoich modułów”. Uwierzcie, że to była straszna groźba, bo muzykiem, jak wspominałem, byłem dupiastym okrutnie. No to się brali...

**Jak się zakończyła działalność magazynu i ile oficjalnie ukazało się jego numerów?**

Kurczę, wiesz że już tak dokładnie nie pamiętam? Myślę, że co najmniej 20 numerów, ja odszedłem bodaj po 18, później FA istniał już w formie czysto elektronicznej (tzn. nie był dystrybuowany na dyskietkach). Jak się zakończyła działalność? Ludzie się powykruszali, Amiga padała, więc ubywało jej posiadaczy, nie było komu pisać, a i redakcja miała już dość, bo sam wiesz, że samodzielne redago-

wanie pisma to „troszkę” męczące i czasochłonne hobby. Właśnie mi się przypomniało, że robiąc pierwsze FA RĘCZNIE wbijałem entery co 16 linii (bo tyle liczyła strona) w CEDzie (edytor) i czasem po kilku godzinach się okazywało, że się machnąłem w zliczaniu linii i strona nie siedzi na ekranie, więc muszę zaczynać od początku. Oj, jaka była radość, jak się okazało, że istnieje coś takiego jak makra w edytorze.

Powiem tak - to była dobra szkoła dzien-nikarstwa.

Naprawdę nie żałuję zarwanych przy tworzeniu FA nocy, a było ich od groma... i wszystko społecznie. Na tworzeniu FA nigdy nie zarobiłem ani grosza. Naprawdę, jak ktoś narzeka na „gieldowych handlarzy”, to niech pamięta też, że ci handlarze w sporej części to byli pasjonaci komputerów, a nie jakieś buraki, co to handlują softem, bo zysk na tym większy niż na, powiedzmy, pietruszce (to tak a propos dość IMO krzywdzącej opinii bodaj Pampama w ostatnim magazynie).

**Po tylu latach miałbyś jeszcze ochotę zająć się pisaniem artykułów do jakiegoś magazynu o tematyce komputerowej?**

Zależy od stawki. :) A serio: mam strasznie mało czasu, ale od czasu do czasu coś tam mógłbym skrobnąć.

**Jak wygląda Twoje obecne życie?**

Zdominowane przez komputery. Praca - wiadomo. Gry, komputery. Moim obecnym wielkim hobby są fotografie cyfrowe, które pracowicie obrabiam na komputerze. A jak mam ochotę się rozerwać - to gram na komputerze. No, troszkę tu przesadzam, gdyż na szczęście istnieje dla mnie życie poza komputerami. Np. uwielbiam czytać! I mam gdzieś e-booki, wolę standardowe książki (aczkolwiek muszę przyznać, że te nowe czytniki czy iPady są całkiem niezłe, na wakacje fajna sprawa, żeby się nie tłuc z plecakiem pełnym lektur). Ogólnie - mało czasu dla siebie, dość sporo stresu, ale przyzwoite pieniądze, które pozwalają mi prowadzić życie na powiedzmy niezłym europejskim standardzie. Ogólnie jednak jestem zadowolony z życia i nie żałuję, że sporą jego część związałem z komputerami. Ja, typowy humanista, który matematykę ma za dzieło szatana... Dziwnie się te ludzkie losy układają. W sumie to już 26. rok mego romansu z komputerem. I mam nadzieję na co najmniej drugie tyle owego owocnego związku.

**Dziękuję za wywiad...**

**wywiad przeprowadził: Ramos**



Niezmiernie cieszy fakt, że został wydany drugi numer Polskiego Pisma Amigowego. Po dokonaniu niezbędnych formalności związanych z zamawianiem gazety przez portal ppa.pl (niestety pismo jest dostępne w sprzedaży wyłącznie przez portal), oczekiwałem z niecierpliwością na przesyłkę.

W tym miejscu warto zwrócić uwagę na fakt, iż od drugiego numeru można nabywać gazetę w formie elektronicznej. Jej koszt jest znacznie mniejszy, a i „wysyłka” łatwiejsza w przypadku, gdy odbiorca nie mieszka w granicach Polski. Wersja ta pojawiła się z niewielkim opóźnieniem w stosunku do wersji papierowej.

Tym razem autorzy uszczęśliwili czytelników dodatkowymi czterema stronami, przy czym cena wzrosła tylko o złotówkę. Gazeta wydawana jest na papierze dobrej jakości i w całości w kolorze. Zarówno ilustracje, jak i tekst stoją na wysokim technicznym poziomie. Muszę jednak stwierdzić, iż strona tytułowa pierwszego numeru o wiele mocniej przypadła mi do gustu. Okładka drugiego numeru nawiązuje do projektu nowej Amigi, która – miejmy nadzieję – niebawem się ukaże (życzę wszystkim Czytelnikom C&A Fan, aby w chwili, gdy czytają te słowa, nowa Amiga była już na rynku). Autorzy, tak jak w poprzednim numerze, apelują do czytelników o artykuły oraz ogłaszają (nieustający - mam nadzieję)

konkurs na najlepszy artykuł. Jest to o wiele lepszy pomysł, niż ogłoszony w poprzednim numerze konkurs na krzyżówkę.

Po porcji Ami-ności możemy zapoznać się z artykułem o porcie TrueCrypt'a pod MorphOSa - czyli o Krypto-sie. Jest to pierwsza część zapowiadanego przez autora cyklu artykułów o tym programie. O tym, że programy takie jak ten są dzisiaj potrzebne, nie trzeba nikogo przekonywać. Dlatego dobrze, że powstaje przewodnik po programie. Zaoszczędzi to z pewnością niektórym osobom wielu nerwów związanych z utratą zaszyfrowanych danych przez nieumiejętne bądź nieostrożne posługiwanie się programem.

Na następnej stronie PekDar kończy swój artykuł z poprzedniego numeru o tym, jak zainstalować AmigaOS 4 na Amidzie 4000. W ramce podano także link do opisu instalacji tego systemu na A1200. Autor pokusił się również o testy czasu wczytywania systemu.

Odwracając stronę znajdziemy krótki wspominkowy artykuł o historii amigowego zlotu, który odbył się w 2002 roku. Wspomina nie kto inny, jak sam Główny Organizator. Opisuje źródła pomysłu, proces organizacji oraz atmosferę tego, bądź co bądź, pierwszego spotkania pod szyldem Amiga Eastern Meeting. Zakończenie artykułu sugeruje, że będziemy mogli jeszcze nieraz powspominać kolejne edycje tego meetingu.

W pierwszej części jednostronicowego artykułu o używaniu FFmpeg'a, Valvit podjął się próby opisanie, czym jest ten pochodzący z Linuksa program, co on robi oraz jak to robi. Artykuł ma charakter wstępu i porządkuje terminologię związaną z obszarem działania programu. Autor obiecuje, iż w następnych częściach dokona dokładniejszego opisu jego poleceń, tak aby wszyscy użytkownicy różnych systemów mogli w pełni korzystać z jego zalet.

Jako że jestem fanem przygotówek, z prawdziwą radością powitałem recenzję znakomitego klasyka, jakim bez wątpienia jest gra Sam & Max Hit the Road. Dzięki emulatorowi ScummVM można znowu pograć w gry „z tamtych lat”. Autor wyjaśnia co uczynić, aby pograć na Eficie z systemem MorphOS. Dalszą część

artykułu stanowi standardowa recenzja: o co chodzi w grze, jak wygląda oprawa graficzna i muzyczna. Ogólnie bardzo fajna recenzja.

W kolejnym artykule pod tytułem „O Eficie w praktyce cz.2” (jak sam tytuł wskazuje, jest to kontynuacja artykułu z poprzedniego numeru) recedent opisuje, jak wygląda sprawa z oprogramowaniem na maszynie wyposażonej w MorphOSa. A nie wygląda wcale źle. Nie ma może wielkiego wyboru, jednak już istniejące programy powinny w zupełności wystarczyć do wszystkich podstawowych aktywności Ami-zjadacza chleba. Autor opisuje, jak sprawa wygląda przy „sieciowaniu”, emulowaniu oraz graniu. Dobrze, że wspomina o sposobach radzenia sobie z softem z Amig klasycznych. Właściwie najczęstszym problemem, na jaki można się natknąć przy używaniu Efiki jest brak wystarczającej ilości pamięci.

Logan w swym artykule podjął się nie lada wyzwania. Próbował przedstawić sytuację wpływającą na dość ponury fakt, iż gry polskich twórców nie stały się przebojami na Zachodzie. Powiem więcej – nikt ich (w większości) nawet za bardzo nie kojarzy. Podszedł do problemu kompleksowo. Wymienił naistotniejsze wady i zgrzyty dosyć chałupniczego sposobu tworzenia i dystrybucji gier w Polsce w latach 90-tych. Artykuł poruszył wszystkie istotne błędy elementów łańcucha developer-klient. Ze swojej strony mogę tylko dodać, że po prostu takie były czasy. Raczkująca gospodarka rynkowa, brak jakiegokolwiek doświadczenia oraz brak funduszy na wsparcie oraz sławna giełda składały się na taką, a nie inną sytuację. My po prostu nie mieliśmy żadnych szans z doświadczonymi w bojach na tym rynku (znacznie zasobniejszym w gotówkę przecież) zachodnimi firmami. Pozostaje cieszyć się, że odcisnęliśmy ślad w Ami-świecie. Poza tym było przecież kilka gier, których nie powstydziliby się żaden wydawca na Zachodzie. Problem z nimi polegał na tym, że dystrybutorzy zachodni mieli dostęp do naszego rynku, a my mogliśmy o dostępie na ich rynki tylko pomarzyć. Może gdyby polscy dystrybutorzy byli bardziej zainteresowani naszą platformą i nie traktowali jej wyłącznie jak wyciskarki kasy... Może wtedy wszystko potoczyłoby się inaczej.

Na kolejnych stronach lord\_spider opisuje proces instalacji Linuksa w dystrybucji OpenSUSE 11 na Pegasosie II. Tekst szczegółowo opisuje cały proces, a dodatkowo wzbogacony jest wieloma mniej lub bardziej czytelnymi zdjęciami.

W artykule „Minimig – powrót do przeszłości”, Emu opisuje swoje boje z tym komputerem. Niestety emulacja pocziwej A500/A600 w postaci małej płytki i to bez obudowy sporo kosztuje. Chyba jest to główna przyczyna braku popularności tego rozwiązania

w naszym kraju. Autor opisał swoje pierwsze wrażenia z kontaktu z zawartością pudełka, która notabene została dokładnie przedstawiona. Scharakteryzował podstawowe i niemałe zresztą możliwości tego „malucha” oraz przedstawił kwestię zgodności oprogramowania z Amig. Artykuł kończy opis możliwości rozbudowy tego urządzenia. Naprawdę zachęcam do zapoznania się z nim, gdyż stanowi bardzo ciekawy opis tego rzadkiego i raczej trudno dostępnego, a jakże ciekawego rozwiązania.

W przedostatnim artykule Pampers prezentuje działanie *MorphOS* 2.4 na Macu mini. Opisano proces instalacji, rejestracji, zalety Ambienta oraz jak system radzi sobie z Internetem.

W ostatnim artykule Adam Zalepa opisał sposób zapisu nut w najpopularniejszych amigowych trackerach, jak *ProTracker*, *OctaMED* i *DigiBooster*. Lektura obowiązkowa dla wszystkich, którzy chcieliby trochę pomuzykować na Amidze.

Strony okładowe wypełnia krzyżówka, rozwiązanie konkursu oraz kolejna część komiksu. W tym numerze akcja powoli zaczyna się rozkręcać i z niecierpliwością czekam na dalszy rozwój wydarzeń. Zależy on tylko od głosów użytkowników portalu ppa.pl, tak więc namawiam do głosowania na ich stronie internetowej.

Zachęcam do lektury pisma. Gołym okiem widać, że redaktorzy poczynili postępy w stosunku do poprzedniego numeru. Cztery dodatkowe strony oraz 11 artykułów na różne tematy, od wspomnień sprzed lat po opisy nowych rozwiązań, zarówno programowych jak i sprzętowych, które nie mają charakteru reklamowego (jak to często w teraźniejszych gazetach bywa) a jedynie prezentują osobiste zdania ich autorów – wszystko to pozwala każdemu amigowcowi na spędzenie kilku miłych chwil z gazetą w ręce. W tym numerze przeważają artykuły o *MorphOS*, ale również użytkownicy Amigi klasycznej znajdą coś dla siebie. Moim zdaniem zdecydowanie za mało jest artykułów o *Amiga OS 4*.

Na koniec tej przydługiej recenzji chciałbym wyrazić nadzieję, że aktualna droga rozwoju zostanie zachowana, czyli każdy następny numer to +4 strony. Oczywiście żartuję. Zdaję sobie sprawę, że ilość artykułów zależy nie tylko od redakcji, ale również od czytelników. Tak więc piszcie, piszcie i jeszcze raz piszcie zarówno do *PPA*, jak i do *C&A Fan*.

Osobiście uważam, że warto było wydać te 20 złotych na gazetę i mam nadzieję, że moja recenzja zachęci do tego również niezdecydowanych amigowców.

Jak dla mnie największą wadą pisma jest zbyt krótki komiks, który powinien gościć na dwóch stronach.

MrMat



W pierwszych dniach października dotarł do mnie najnowszy trzeci już numer Polskiego Pisma Amigowego. Po odpowiedzi na pytanie czy i tym razem warto było wydać dwie dychy na gazetę zapraszam do recenzji.

Ogólny poziom pisma jest niezmienny, czyli kolorowo na dobrym papierze i czytelnie. Tak trzymać chłopaki! Rozbawiła mnie nieco okładka pisma. Gdybym nie wiedział czego dotyczy to pomyślałbym, że głównym tematem pisma są problemy z odpadami elektronicznymi, głównie ze „starych” komputerów. Oczywiście na drugiej stronie znalazł się nieśmiertelny apel i konkurs dla czytelników o nowe artykuły. Nagrody są całkiem przyzwoite. My do apelu jak zwykle się oczywiście dołączamy. Piszcie do *PPA* i *C&A Fan* jak najwięcej, jeżeli chcecie aby ukazywały się w miarę regularnie.

Po porcji newsów możemy zapoznać się ze zdaniem Adama Zalepy na temat społeczności amigowej w dzisiejszych czasach. Autor wychodząc od historycznych czasów triumfu Amigi dochodzi do współczesnych podziałów i analizuje ich przyczyny. Bardzo ważny postulat wysunięty w artykule każe każdemu komu leży na sercu dobro Amigi zastanowić się nad przyszłością; co z tego wyniknie – czy środowisko będzie tonęło w coraz większym zacietrzewieniu nad swoimi racjami czy zacznie dzielić się doświadczeniami i powoli odbudowywać pozycję Amigi.

Na kolejnych dwóch stronach możemy zapoznać się z drugą częścią artykułu o *Kryptosie* autorstwa Grzegorza Kraszewskiego. Autor opisuje sposób podawania klucza oraz hasła wraz z montowaniem zakodowanej partycji w systemie. Artykuł kończy pobieżny przegląd

# NASZA RECENZJA

atakowna *TrueCrypta*. Okazuje się, że program nie zapewnia 100% bezpieczeństwa, ale który zapewnia? Problemy wynikają jednakże bardziej ze sposobu działania elektroniki komputera niż samego softu. Autor podaje linki do szczegółowych materiałów o sposobach radzenia sobie z *TrueCryptem*. Jest to o tyle ważne, iż używając różnych zabezpieczeń trzeba mieć zawsze świadomość ich ograniczeń.

Po „ciężkim” artykule na ważny temat, docieramy do znacznie „lżejszej” recenzji Marcina Skawińskiego. Okazuje się że nawet dzisiaj można kupić grę na Amigę (i to w wariantach na wiele systemów) tak jak za dawnych czasów, czyli w pudełku i bez instrukcji, he,he. Ale kto by się takimi szczegółami przejmował jeżeli chodzi o nową grę na nowe systemy i tak prostą jak Amijeweled. Autor dzieli się z czytelnikami swoimi spostrzeżeniami z wersji na *Amiga OS 4.1*. Ogólnie jest to bardzo fajny klon znanego *Bejeweleda*.

W tym numerze rozpoczęty został nowy cykl wywiadów pod tytułem 12 pytań do... Pierwszy wywiad został przeprowadzony z legendarnym *Lazurem*. Mam nadzieję, że autorzy trochę zmodyfikują formułę. Chodzi mi o te 12 pytań. Jeżeli wywiad fajnie się rozkręca to należy go kontynuować, aż do wyczerpania tematów, a nie do 12 pytań. Z kolei jeżeli ktoś nie ma nic ciekawego do powiedzenia to po co go męczyć (no powiedz jeszcze coś..., i jeszcze tylko 2 pytania i dam Ci spokój). Pomysł z wywiadami przeprowadzanymi przez Logana jest znakomity. Facet zna się na rzeczy. Ale formuła powinna być nieco zmodyfikowana.

Na następnej stronie możemy zapoznać się z kontynuacją opisu działania *MorphOS 2.4* na *Macu Mini* autorstwa Wiktora Głowackiego. W artykule zaprezentowano działanie stosu USB oraz gier z różnych kategorii działających pod tym systemem. Jeżeli ktoś po graniu zechce stworzyć Amidzielo, może odpalić najbardziej znane z Amigi klasycznej programy muzyczne i graficzne. Autor opisuje także jakich narzędzi użyć, aby po ciężkiej pracy nad tworzeniem Amidziela zrelaksować się przy filmie. W dalszej

części opisano wady i niedoróbki systemu, co sprawia, iż artykuł nie jest tylko peanem na cześć systemu, ale rzetelnym opisem jego wad i zalet.

Tak oto dochodzimy do *Historii Amiga Eastern Meetingu* oczami bywalców, Flei, il-capitano, Drako i ZED'a. Opisują oni wyjątkowy klimat imprezy i zachęcają do wspólnego budowania go w przyszłości.

Na kolejnych stronach Adam Zalepa opisuje system do nieliniowej edycji obrazu, który zebrał w swoim czasie bardzo pochlebne opinie wśród prasy video. Chodzi o system *Casablanka 1*. Nawet *PC World Komputer* w 2002 roku zamieścił bardzo pozytywną recenzję tego produktu. Okazuje się, że po rozebraniu urządzenia mamy do czynienia ze zwykłą Amigą klasyczną wzbogaconą o dodatkowy osprzęt i specjalnie dostosowane oprogramowanie dostępne również na Amigę „domową”. Ciekawe gdyby recenzja dotyczyła Amigi w standardowej obudowie, czy była by też taka entuzjastyczna. Czy nie było by konkluzji że to może i fajny sprzęt ale taki „przestarzały” i nie oparty o jedynie słuszny hardware i system. Autor opisuje zalety tego systemu, jak to wszystko działa w praktyce oraz podaje jego parametry techniczne.

Kolejny tytuł tego autora, kontynuuje cykl o trackerach amigowych. W tym odcinku opisał sposób działania poleceń A oraz C. Wszystkich którzy chcieliby popробować sił w Amimuzykowaniu zachęcam do przeczytania. Wiedza zdobyta na takim „kursie” z pewnością przyda się nie tylko w Amiświecie.

Kolejny artykuł to recenzja... Tak zgadliście naszego poprzedniego numeru. Dzięki chłopaki.

Dochodzimy do zwycięzcy konkursu z drugiej strony *PPA*. Jest to art o sterowaniu PCtem za pomocą Amigi. Dzięki usłudze zdalnego pulpitu można zaoszczędzić sporo miejsca na biurku na dodatkowym monitorze. Ale w zamian trzeba wiedzieć jak to wszystko po-ustawiać. Z artykułem MarX'a gwarantuję, że nie będzie z tym żadnego problemu. Naprawdę

zasłużony zwycięzca.

Alekc w swym artykule opisuje sposoby przeniesienia i modyfikacji algorytmu *MD5* pod system *AOS4.1*. Powiem tylko, że algorytm ten służy do testowania integralności archiwów. Po bliższe informacje odsyłam do artykułu.

Grzegorz Kraszewski ułożył tutorial o cieniowaniu ikon w *MorpOS*. Okazuje się, że można tego dokonać za pomocą posiadającej dodatkowe możliwości przeglądarki obrazków. Stanowi on dokładny przepis jak, krok po kroku, wykonać profesjonalnie wyglądający cień pod własną ikonę. Naprawdę bardzo ciekawy art. Opisane tu kroki są tak naprawdę uniwersalne i można je po odpowiedniej modyfikacji zastosować do dowolnych elementów do których potrzebujemy „dorobić” cień.

Jeżeli ktoś z Was zgromadził potężną bazę MP3 i nie może nad nimi zapanować to po przeczytaniu artykułu o *AmiTunes* Sebastiana Rosy nie powinien mieć z tym większego problemu. Jest to typowa recenzja programu do tego celu działającego na klasycznych Amigach. Opisana wersja 0.6 sugeruje, że program dopiero nabiera wiatru w skrzydła i za jakiś czas może to być bardzo solidne narzędzie. Tym bardziej już teraz powinno się go zacząć poznawać.

I tak oto nieubłagane dochodzimy do ostatniego artykułu. Są to publicystyczne rozważania MarX'a na temat dzisiejszego Amigowania. Opisuje podziały, ale także dostrzega w nich wspólnego Amigowego ducha który „ponad podziałami” jednoczy ludzi związanych z tym komputerem i nie pozwala mu umrzeć.

Na przedostatniej stronie znajduje się krzyżówka i inne teksty z nią związane. Szybki rzut okiem na zawsze za krótki komiks i zadajemy sobie sprawę że oto znów dojrżeliśmy do końca i trzeba będzie czekać kolejne trzy miesiące do ukazania się kolejnego numeru. Mam nadzieję, że ze względu na zbliżające się ochłodzenie klimatu i skrócenie naświetlania światłem słonecznym numer ukaże się przed czasem.

Podsumowując chłopcy z *ppa.pl* mają za sobą kolejny udany numer czasopisma i warto wydać na niego te dwie dychy, aby potrzymać kawałek dobrej papierowej amiprasy w ręku i znów poczuć klimat jak za dawnych lat.

MrMat

# Pograjmy jak za dawnych lat...

## ROK 1984 - cz.2

Dzisiejszy artykuł rozpocznę od omówienia gry, którą znają chyba wszyscy, chociaż pewnie nikt nie umieściłby jej nawet w szerokim gronie swoich ulubionych tytułów ze świata C-64. Ponadto produkcja ta pasuje jak ulał do fali upałów, które w momencie pisania tego tekstu nawiedzają całą Polskę. Mowa o **Strip Poker: A Sizzling Game of Chance**.



Pierwotnie, w 1982 roku, ukazała się ona na komputerach Apple II. W kolejnych latach, w grę można było zagrać również na C-64, Plus/4, Amidze, różnorodnych Atarynkach, a także na PC-cie. Trudno nazwać Strip Poker pozycją ważną, czy też po prostu dobrą, niemniej gdyby potraktować ją jako pewnego rodzaju zjawisko socjologiczne, wydaje się, iż nie może dla tego tytułu zabraknąć miejsca w serii artykułów o roku 1984. Reguły gry zna ją chyba wszyscy. Naszą pokerową przeciwniczką jest jedna z pięknych (?) 8-bitowych panienek. Gdy czyjeś saldo spadnie do zera, za pożyczoną setkę oferuje ona kolejne części swojego ubioru. Gra kończy się, gdy któryś z graczy pozostanie tak, jak go Pan Bóg stworzył. Oryginalnie Strip Poker oferował rozgrywkę z dwiema Paniami, dzięki dodatkowym dyskom z danymi wybór stał się większy. Przy tego typu produkcjach zwraca się uwagę na dwie rzeczy: inteligencję komputerowego przeciwnika i... grafikę, rzecz jasna. O tej drugiej nie ma co obecnie nawet wspominać. Jeśli potencjał erotyczny był w tej grze kiedykolwiek obecny, to już dawno wywietrzył. Z inteligencją komputera nie jest najgorzej, aczkolwiek po kilku partyjkach można zauważyć pewne prawidłowości, które ułatwiają podejmowanie decyzji, chociażby przy wymianie kart. Wydaje mi się, iż w latach 80-tych powstanie takiej gry miało pewien głębszy sens. Współcześnie, produkcje tego typu nie dostarczają chyba nikomu takich emocji; uj-

mując to w bardziej bezpośrednie słowa: goliżna spowszedniała. Pamiętam, jak z kolegami wymieniałem się historiami typu „co zrobiłeś, gdy podczas gry w Strip Pokera weszła twoja mama”. To se ne vrati. Dzisiaj Strip Poker firmy Artworx pozostaje raczej historyczną ciekawostką. Jeśli ktoś pragnie zagrać w rozbieranego pokera, bez problemu znajdzie współczesne produkcje. Z kolei ci, którzy koniecznie chcą porozbierać 8-bitowe panienki, niech lepiej sięgną po, chyba nieco lepszą grę, o tytule Samantha Fox Strip Poker (Martech, 1986). Tylko mała uwaga: upały mogą przed domownikami usprawiedliwić brak jedynie niektórych części stroju :)

Przejdźmy do obiecanych, w poprzedniej części artykułu, strzelanek. Jednym z tytułów wydanych w 1983 roku, z którego opisu w artykule o tymże roku musiałem z ciężkim sercem zrezygnować, był Falcon Patrol. Ale nic straconego – rok później ukazała się druga część tej gry, która na dodatek na pierwszy rzut oka wygląda niemalże identycznie, jak poprzedniczka. Tym samym w **Falcon Patrol II** ponownie siadamy za sterami myśliwca, mając za zadanie zestrzelić wszystkie wrogie jednostki.



Podobnie jak w części pierwszej można latać w lewo lub prawo (obszar gry „zapętla się”), zerkając jednocześnie na radar w poszukiwaniu wrogich maszyn. Tak samo jak w poprzedniczce, należy pilnować poziomu paliwa, a także liczby pozostałych rakiet, a wszelkie braki uzupełniać na rozmieszczonych tu i ówdzie lądowiskach. Zmienił się nieco krajobraz – bazę wojskową z pierwszej części zastąpiły piramidy i budujące świetny klimat zniszczone budynki. Wprowadzono czytelny podział na poziomy trudności (do kolejnego etapu przechodzimy niszcząc odpowiednią

liczbę wrogich jednostek), zachowując jednocześnie ciągłość gry znaną z części pierwszej. Zmieniono wrogie maszyny – tym razem gracz musi sprostać trzem rodzajom helikopterów, różniących się od siebie pod względem sprawowanej funkcji. Nowością w **Falcon Patrol II** jest możliwość oddawania strzałów ukośnie w kierunku ziemi, mogąc tym samym niszczyć różne naziemne działka oraz lądowiska. Na szczęście zachowano, znane z pierwszej części gry, niesamowicie klimatyczne efekty dźwiękowe. Śmiem twierdzić, iż odgłosy wydawane przez samolot gracza to najpiękniejszy szum ze świata C-64. Jeden z użytkowników portalu lemon64.com napisał, iż „choć to zły sequel, to wciąż dobra gra”. I jestem w stanie taką opinię zrozumieć – **Falcon Patrol II** wbrew pozorom istotnie różni się od poprzedniczki. O ile w pierwszej części należało urządzać swoiste polowanie na wrogie samoloty, tym samym wpierw je goniąc, w drugiej części z dogonieniem wolnych śmigłowców nie ma problemów – większy nacisk położono nie na lot, a na strzelanie. Co kto woli... Chociaż osobiście uważam, że warto zagrać w obie części gry, nieco wyżej oceniam właśnie **Falcon Patrol II** – ładniej wyglądającą i bardziej dynamiczną od poprzedniczki produkcją.

Wyżej wspomniany użytkownik lemon64.com zauważył również, iż **Falcon Patrol II** przypomina nieco klon **Defendera**, starszej gry z automatów, której wersja na C-64 ukazała się w 1983 roku. W tym miejscu warto chociażby wspomnieć o **Dropzone**, przez wielu uznawanej za najlepszą grę tego typu na C-64, do której zwolenników – ze względu na zbyt duży, moim zdaniem, poziom trudności – jednakże nie należę.



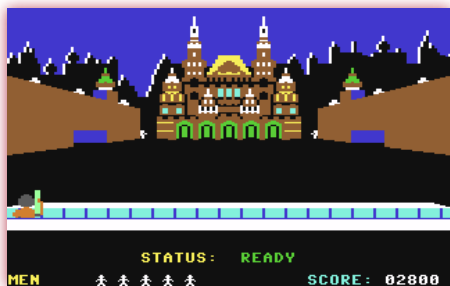
Do grupy „wielkich nieobecnych” w tym

artykule dołączam również Zaxxon, dosyć wolną strzelankę w rzucie izometrycznym, a także jedną z trzech gier wartych wspomnienia z wyrazem „raid” w tytule – **Raid on Bungeling Bay**, w której za pomocą helikoptera dokonujemy nalotów na terytorium wroga, niszcząc różnorakie pojazdy czy strategiczne obiekty.



Jest wysoce prawdopodobne, że kiedyś wrócę do tego tytułu w osobnym artykule. Póki co przejdźmy do kolejnego raidu... to jest, do kolejnej gry.

Pamiętacie Bruce'a Carvera? To właśnie on, w 1983 roku, popisał się kapitalną grą **Beach Head**. W 1984 roku mocniej zaznaczył on swoją obecność w świecie C-64, tym razem dzięki jeszcze lepszej produkcji pod tytułem **Raid Over Moscow**, która nie tylko jest coraz to bardziej odległym wspomnieniem ery C-64, ale również – ośmiobitowym śladem po zimnej wojnie. Jej celem jest bowiem powstrzymanie trzech ataków nuklearnych ZSRR na USA i zniszczenie całej infrastruktury pozwalającej na tego typu akcje, włącznie z głównym reaktorem znajdującym się – a jakże – w Moskwie.



Wydany na różnych ośmiobitowych platformach tytuł, podobnie jak wspomniany już **Beach Head**, składa się z kilku mini-gier. W pierwszej kolejności musimy wyprowadzić samolot z hangaru, co ze względu na specyficzne sterowanie nie jest rzeczą prostą. Tu cenna informacja: drzwi hangaru otwieramy klawiszem F7. Z kolei przełączenie mapa-hangar odbywa się za pomocą spacji. Następnie, mając przed sobą kawałek globu z zaznaczonymi miastami USA i ZSRR, kierujemy samolot w stronę tej miejscowości, z której właśnie nastąpiło wystrzelenie rakiety z ładunkiem atomowym. Tu następuje etap,

któremu najbliższej do tradycyjnej strzelanki w rzucie izometrycznym. Po przelocie nad całą bazą wojskową musimy już tylko zniszczyć ostrzeliwujące nas silosy, po czym dane centrum sterujące atakiem można uznać za zniszczone. Na całe zadanie mamy ograniczoną ilość czasu, który zależy nie tylko od wybranego na początku gry poziomu trudności, ale również miasta, z którego została wystrzelona raketa oraz celu ataku. Po trzykrotnym wykonaniu tego zadania, samolot ląduje w Moskwie, gdzie następuje chyba najciekawsza część gry. Wyposażeni w bazookę, ostrzeliwując snajperów i czołgi, musimy przedrzeć się do wnętrza Kremla, gdzie za jednym z pięciorga drzwi (po trafieniu w które zmieniają one kolor na biały) znajduje się główny reaktor. Z kolei, aby zniszczyć ów reaktor, musimy kilkakrotnie rzucić dyskiem tak, by odbity od ściany przedmiot trafił w tył poruszającego się robocika. Po zainkasowaniu kilku uderzeń gra wciąż nie jest skończona – reaktor staje się niestabilny i mamy tylko dwie minuty na jego zniszczenie. Tym samym **Raid Over Moscow** oferuje 2 różne sekwencje końcowe, w zależności od tego, czy uda nam się ze zniszczonej Moskwy uciec czy też nie. Mogę tylko się domyslać, jaką radochę mieli gracze z krajów tzw. demokracji ludowej, dokonując małej rozwalki na Kremlu. Gra miała swój wydźwięk również w popkulturze – odgrywa ponoć istotną rolę w książce „Extraleben” Constantina Giliesa. Ciekawostką jest fakt, iż **Raid Over Moscow** jest szczególnie znana w... Finlandii. Jeden z tamtejszych komunistów wszczął parlamentarną dyskusję na temat tego, czy taka gra powinna była w ogóle trafić na tamtejszy rynek. Oczywiście tym sposobem spowodował tylko dodatkowy rozgłos i lepszą sprzedaż produkcji. Z kolei w Polsce mocno krytycznie odniósł się do gry magazyn Komputer. No cóż, przecież **Raid Over Moscow** obrażał miłujących pokój Sowieć. A jeśli chodzi o Bruce'a Carvera... mogę tylko obiecać, że w tym cyklu jego nazwisko pojawi się jeszcze co najmniej dwukrotnie.

Sporo było tych „raidów” w roku 1984, pora przejść do kolejnej, najprawdopodobniej najsłynniejszej gry zawierającej ten wyraz. Mowa oczywiście o **River Raid**. Do dziś pamiętam pewne stanowisko komputerowe tuż przy wejściu do szatni basenu MOSiRu, na którym można było zagrać właśnie w tą grę. Mój kolega wziął joya w ręce, wykręcając absolutnie fenomenalny i nieosiągalny dla mnie rekord. Wspominam o tym nie po to, by po raz już chyba kolejny napisać, że w strzelankach to ja mocny nie jestem, ale dlatego, by pokazać, że skoro ten tytuł dotarł nawet w takie miejsce, jak MOSiR-owy basen, to tym bardziej nie ma chyba miłośnika 8-bitowych gier, który by się z **River Raid** nie ze-

tknął. Produkcja została wydana w 1982 roku przez firmę Activision na konsolę Atari 2600, wersji na C-64 doczekała się dwa lata później. Z kolei wydany w 1988 sequel swojej odsłony na C-64 się nie doczekał.



Pisanie na czym polega gra, to coś z kategorii „oczywista oczywistość”, ale co tam: w **River Raid** siadamy za sterami samolotu lecącego nad rzeką. Musimy zestrzeliwać wrogie samoloty, balony, czołgi czy też statki, uzupełniając niedobory paliwa na platformach z napisem „fuel”. Poziomy wyznaczone są przez kolejne mosty, które oczywiście również należy zniszczyć. Rzecz jasna, że im dalej, tym trudniej – przeciwników przybywa, zaś teren staje się bardziej zdradliwy. Z powodu swojej prostoty **River Raid** może dziś zarówno wywołać kpiący uśmieszek, jak i wciągnąć, jak za dawnych lat. Wszystko zależy od nastawienia. Wyłącznie natomiast w kategorii „kiedyś to były czasy” należy ocenić kontrowersje, jakie ta produkcja wywołała w Niemczech. Otóż **River Raid** został umieszczony przez pewną organizację cenzorską na liście gier zakazanych. W oświadczeniu uzasadniającym taką decyzję napisano m.in., iż „niepełnoletni muszą wcielić się w rolę bezkompromisowego żołnierza i agenta od destrukcji”. Biedna Carol Shaw (która **River Raid** stworzyła) zapewne nie przypuszczała, jaka bestia wyszła spod jej programistycznych palców. Cóż, na całe szczęście dziś jesteśmy już pełnoletni, czyż nie?

Ci, którzy powyższe strzelanki uważają za zbyt – nomen omen – przyziemne, być może chętniej przypomną sobie dwa inne tytuły. Na początek wzniesmy się tylko nieco powyżej ziemskiej atmosfery. To właśnie tutaj rozgrywa się akcja **Beamrider**.



Pierwotnie, stworzona na konsolę Intellivision gra firmy Activision (tak, znowu oni)

doczekała się swoich wersji na większość maszyn 8-bitowych, w tym oczywiście również na C-64. Fabuła? Kosmici zamknęli Ziemię w dziwnym polu siłowym świecących promieni. Pilotując tytułowego „jeźdźcę promieni” musimy oczyścić 99 sektorów tego wrogiego pola. Broń? Proste działko laserowe z nieograniczoną ilością amunicji, a także skończona liczba torped. Aby oczyścić dany sektor, należy zniszczyć 15 białych statków, unikając przy tym innych, z poziomu na poziom coraz liczniejszych, przeszkadzajek. Pod koniec każdej rundy mamy możliwość zniszczenia dużego statku-matki (za pomocą torpedy), by zdobyć dodatkowe punkty. Trzeba przyznać, iż **Beamrider** to interesująca i oryginalna wariacja tematu „kosmiczna strzelanka”. Ot, chociażby ze względu na swoją symulowaną trójwymiarowość. Albo ze względu na fakt, iż pilotowany przez gracza statek może znajdować się tylko na jednym z pięciu promieni. To z kolei każe mi przypuszczać, iż ta właśnie gra mogła zainspirować ekipę z Hewson przy okazji tworzenia świetnej strzelanki z 1988 roku o tytule **Eliminator**. Pewne natomiast jest to, że **Beamrider** jest dziś dostępny nie tylko na PC-tach, lecz również konsoli Xbox 360.

Przenieśmy się jeszcze dalej, gdzieś między zdegradowanego do roli planety karłowatej Plutona, a Neptuna – ósmą planetę Układu Słonecznego. To właśnie tutaj rozpoczyna się akcja kolejnego hitu rodem z automatów, wydanej pierwotnie przez Konami w 1983 roku gry **Gyruss**. Jej celem jest dotarcie do Ziemi i oczyszczenie Układu Słonecznego z kosmicznych najeźdźców. Tu chciałoby się napisać, iż zadanie nie będzie proste z takich czy innych powodów, ale... tak nie jest.

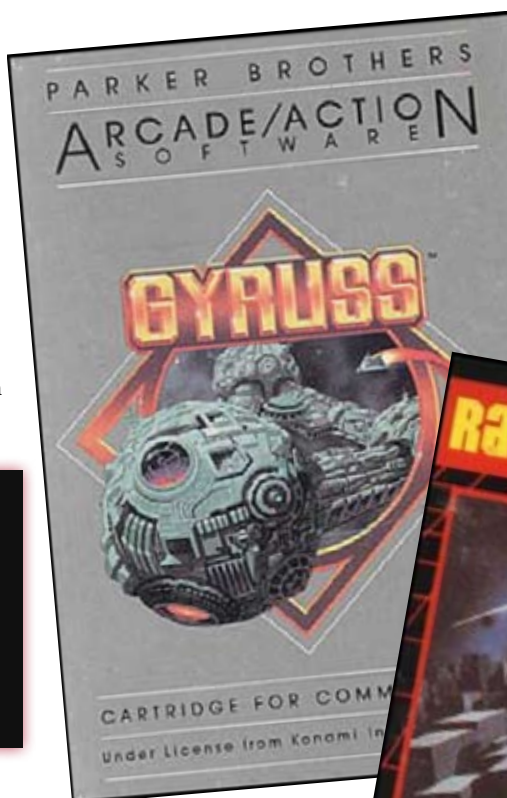


**Gyruss** jest bowiem raczej prostą strzelanką, jedną z nielicznych, którą udało mi się bez żadnych oszustw ukończyć. Lecz chyba nie tylko dlatego to moja ulubiona gra z tego gatunku. Po pierwsze – podoba mi się umieszczenie akcji w Układzie Słonecznym. Po drugie – niezwykle oryginalnie rozwiązano przemieszczanie się po polu gry. Otóż autor, Yoshiko Okamoto (w późniejszych latach podpisał się m.in. pod 1942) stworzył takie wrażenie, jak gdyby sterowany przez gracza statek kosmiczny leciał w głąb ekranu. Tym

samym wszelkie wrogie obiekty – inne statki, asteroidy czy satelity – wylatują właśnie ze środka ekranu, zaś pilotowany pojazd może wokół nich krążyć. Jednakże gra stała się słynna, chyba przede wszystkim, ze względu na ścieżkę dźwiękową. Nie tylko bowiem **Gyruss** był pierwszą produkcją posiadającą muzykę w stereo, ale też wykorzystano tu kompozycje Jana Sebastiana Bacha – konkretnie Toccata i Fugę w d-moll. I chyba to wszystko razem powoduje, iż tak bardzo lubię tą grę. Podróż przez Układ Słoneczny, przy takiej, a nie innej muzyce nastraja mnie jakoś nostalgicznie. Ostatecznie, jak śpiewała Joni Mitchell, jesteśmy gwiazdny pyłem... Wychoząc z okowów metafizyki dodam jeszcze, iż gra jest dostępna również na współczesnych konsolach do gier, jak PlayStation czy Xbox 360.

Na dzisiaj to tyle. Jako że temat strzelanek okazał się być bardzo zajmującym, obiecany komnatówkami – i nie tylko – już na 100% zajmę się w następnej części artykułu.

p.a.





# O przerwaniach i cyklowaniu na poważnie

Na pewno nie jeden z was, wiernych czytelników C&A, pamięta prowadzone kiedyś na łamach gazety kursy, od których można było dostać wypieków na twarzy oraz na które po prostu czekało się z miesiąca na miesiąc. Mowa tu o „Jak napisać własne Demo”, albo „Pamiętnik artylerzysty” skupiające się na robieniu rzeczy pozornie niemożliwych, które wyciskały siódme poty z pocziwego C64.

Istotnym elementem przy każdej z prezentowanych wówczas „sztuczek”, było śledzenie położenia tzw. rastra, i napisanie programu tak, aby kontrolował w pełni to, co dzieje się na ekranie monitora. Nie będę tutaj rozwodził się nad techniczną stroną uzyskiwania obrazu, albo co się kryje w słowie **cyklowanie**. W tamtych archiwalnych kursach było to już dobrze opisane, a i dzisiaj również ta wiedza nie jest tajemna i łatwo ją odnaleźć w Internecie. Hmm... to, po co właściwie zaczynać pisać o czymś co już było i jest dobrze znane? Właśnie w tym problem, że przeglądając po raz kolejny te kursy i artykuły, zdałem sobie sprawę, o niepełnej wiedzy przekazanej w tamtym okresie. Każdy z opisywanych efektów, oczywiście działał, ale oprócz części programu odpowiadającej za sam efekt, istniała część, która przygotowywała środowisko komputera do jego uruchomienia. Ważne są również zasady przy budowaniu takich programów w oparciu o specyficzne własności maszyny, które wraz z konfiguracją środowiska były niestety zaniedbane. Teksty w stylu „...jeśli linia się szarpie, dostawcie klika NOP-ów przed, albo po, to pomoże, nie pytajcie dlaczego...” świadczą też o niepełnej wiedzy autora, o zjawiskach zachodzących w C64. Oczywiście nie mam zamiaru nikogo krytykować, czasy były jakie były, potrzebna wiedza nie była dostępna, a ja sam raczkowałem w tych sprawach – ucząc się właśnie z tych kursów.

Dziś, minęło kilkanaście lat, i „dużo wody upłynęło” :-), myślę jednak, że warto nadrobić te zaległości, aby pisanie takich programów nie przysparzało niepotrzebnych frustracji, a świadomość pełnej kontroli nad tym co robimy dodawała jeszcze większej satysfakcji w kodowaniu.

Traktujmy ten artykuł jako uzupełnienie pewnych podstaw, więc chcąc nie chcąc trzeba od nich jednak zacząć. Co oferuje nam C64? Przerwania (interrupt) – czyli takie impulsy, które informują procesor, że coś się stało i przerywamy robotę, bo trzeba obsłużyć to zdarzenie. W praktyce oznacza możliwość chwilowego porzucenia wykonywania jednego programu, i uruchomienia drugiego, który po zakończeniu, pozwoli wrócić do pierwszego i kontynuować jego działanie. Wyróżniamy dwa typy przerw: *maskowalne (IRQ)* i *niemaskowalne (NMI)* – krótko: jedno można wstrzymywać, drugie nie. Oba można uruchamiać i zatrzymywać – wyjątek stanowią niemaskowalne sprzężone fizycznie do procesora: klawisz RESTORE, złącze CARTRIDGE. W tych nic nie wskóramy kodem (przynajmniej teoretycznie) i nie będziemy się nimi zajmować. Maskowalne też dzielimy, tu ze względu na źródło pochodzenia, a może to być zdarzenie na grafice, albo nie – to „nie” dotyczy generowania przez liczniki odmierzające czas (*timer-y*). I jeszcze zdarzenia na grafice dzielimy na np. sprite „dotknął” sprite, albo grafiki, albo te które nas najbardziej interesuje – gdy wyrysowywanie ekranu osiągnęło interesujący nas numer linii poziomej (rastra). Oczywiście żeby takie zdarzenia nastąpiły, przedtem trzeba je odpowiednio zdefiniować. Otojednoznacznych ustawień przerwania:

```
SEI
LDA #<IRQ
```

```
STA $0314
LDA #>IRQ
STA $0315
CLI
...

IRQ
DEC $D021
INC $D019
JMP $EA31
```

I co my tu mamy, w zasadzie nic... wszystko stare i dobrze znane – przerwanie oczywiście wykona się (kolor tła zacznie migotać), ale w którym miejscu dokładnie będzie ta zmiana to nie wiadomo - proponuję w emulatorze ustawić szybkość na 1%, wtedy będzie to można gołym okiem zaobserwować. Spróbujmy jednak dokładnie przeanalizować, co się stanie po wykonaniu powyższych instrukcji. SEI wyłączenie przerw (tych które są maskowalne), znaczy tylko tyle że podczas ustawiania naszego przerwania, jakieś inne bieżące, nie zaskoczy nas i nie przejmie kontroli w trakcie, aż do komendy CLI. Co prawda wykonanie tych kilku instrukcji to bardzo mały odcinek czasu dla procesora i prawdopodobieństwo takiego zdarzenia jest nikłe, ale licho nie śpi. Dalej jest ustawienie wektora przerw – czyli adresu w pamięci z naszym programem, który ma się wykonać w przerwaniu IRQ, najpierw młodszy bajt adresu do \$0314, a potem starszy do \$0315. W sumie dlaczego jest to ujęte w strefę ochronną od przerw? Że niby, przerwie jakąś instrukcję w połowie i komputer się zawiesi? O to akurat nie musimy się martwić, konstrukcja wykonywania przerw przez inżynierów C64 na to nie pozwala – w chwili tego zdarzenia, aktualnie wykonywany rozkaz będzie dokończony, a dopiero potem przerwanie zacznie „swoje”. No to, po co? Bo może nastąpić nieszczęśliwy zbieg okoliczności i przerwanie

zaatakuję tuż za *STA \$0314*, i co wtedy? Nie-  
dobrze, młodszy bajt został wpisany, a starszy  
nie – czyli? Czyli Starszy bajt został z adresu  
bieżącego przerwania – rezultat: niewiadomy  
– przerwanie (które akurat przerwało nasz  
program) „poleci” gdzieś w nieokreślony ad-  
res w pamięci no i tylko zależy jakie spotka  
tam instrukcje, ale zazwyczaj skutki są opła-  
kane, włącznie z zawieszeniem komputera. Po  
wpisie starszego bajtu mamy *CLI*, przerwanie  
mamy ustawione. Co dalej? – adres przerwania  
jest skierowany na naszą procedurę *IRQ*, ale  
kiedy tam trafi? Nie wiadomo – można tylko  
przypuszczać, że jeśli komputer jest świeżo po  
resecie, a inne programy nie poprzestawały  
rejestrów to standardowo przerwanie ma-  
skowalne, do którego się dokleiliśmy (*\$0314*,  
*\$0315*) - a co ciekawe wcale nie jest to te ge-  
nerowane przez rejestr rastra (*\$D012*), tyl-  
ko przez układ CIA (czyli licznik) co jakieś  
16500 cykli zegara (ok. 260 linii rastra) w wer-  
sji PAL, a więc praktycznie za każdym razem  
w innym miejscu. Nie o to nam chodziło...  
Takie wykorzystanie przerwania, najczęściej  
jest stosowane, gdy mamy do czynienia z Ba-  
sicem – nie wymaga ścisłej synchronizacji,  
a coś ma się wykonywać w tle niezależnie od  
programu w Basicu (np. odgrywanie muzyki).  
W zasadzie jest tylko takim nieśmiałym przy-  
klejeniem się do systemu (Kernal, Basic), który  
cały czas robi swoje, ale już odrobinę wolniej –  
w każdym razie zupełnie nie dba o nasz efekt.  
Jeśli chcemy wyświetlić fajne poziome li-  
nie, albo zwiększyć liczbę sprite, albo cho-  
ciaż zrobić płynny przesuw czegokolwiek,  
to musimy bardziej wgryźć się w system  
i przykręcić mu trochę przysłowiową „śrubkę”.  
Spróbujmy lepiej sprecyzować wywołanie  
przerwania.

```
SEI
LDA #<IRQ
STA $0314
LDA #>IRQ
STA $0315
lda #$7f
sta $dc0d
sta $dd0d
    lda $dc0d
lda $dd0d
lda #$01
sta $d01a
lda #$xx
sta $d012
    LDA $D011
AND #$7f
STA $d011
lda #$ff
sta $d019
cli
...
```

IRQ  
DEC \$D021  
INC \$D019  
JMP \$EA31

Początek w zasadzie bez zmian, „dokleja-  
my” nasze *IRQ* w *\$0314/\$0315*, a potem jest  
wpis do dwóch rejestrów CIA odpowiedzial-  
nych za przerwanie licznikowe (albo inaczej  
zegarowe). Żeby wszystko było jasne, podaję  
dokładną „rozkładkę” tych rejestrów:

| \$DC0C Interrupt (IRQ) Control Register<br>[def. \$81 (%10000001)] |   |
|--|---|
| Bit 7  | On Read: 1 = Interrupt occurred<br>On Write: 1 = Set Int.-Flags, 0 = Clear Int.-Flags<br>FLAG1 IRQ (Cassette Read / Serial Bus SRQ Input) |
| Bit 4  | Serial Port Interrupt (\$DC0C full/empty)   |
| Bit 3  | Time-of-Day Clock Alarm Interrupt   |
| Bit 2  | Timer B Interrupt (Tape, Serial Port)   |
| Bit 1  | Timer A Interrupt (Kernal-IRQ, Tape)  |
| Bit 0  |   |

| \$DD0D Interrupt (NMI) Control Register<br>[def. \$7F (%01111111)] |   |
|--|---|
| Bit 7  | On Read: 1 = Interrupt occurred<br>On Write: 1 = Set Int.-Flags, 0 = Clear Int.-Flags<br>FLAG1 NMI (User/RS232 Received Data Input) |
| Bit 4  | Serial Port Interrupt (\$DD0C full/empty)   |
| Bit 3  | Time-of-Day Clock Alarm Interrupt   |
| Bit 2  | Timer B Interrupt (RS232)   |
| Bit 1  | Timer A Interrupt (RS232)   |
| Bit 0  |   |

Czyli wpisując tam wartość *\$7F* wyłączamy  
wszystkie rodzaje przerwania (jakie się da)  
wywołane przez oba układy CIA. Tu trzeba  
zauważyć pewną właściwość – wyłączamy  
poprzez zapalenie odpowiedniego bitu,  
a jedynie znacznik (flaga) wystąpienia które-  
gokolwiek z wyłączanych przerwania rejestru  
(bit7) zerujemy. To jeszcze mało – jak widać  
z opisu, Bit7 ma jakby dwie oddzielne funk-  
cje, inne przy zapisie, a inne przy odczycie.  
To, że go zapalimy czy zgasimy, nie ozna-  
cza wcale że po odczycie da nam ostatnią  
wartość - jest kilka rejestrów w c64 które  
działają w ten sposób. Dlatego też po wpi-  
sie do tych rejestrów, następuje zaraz ich  
odczyt (*LDA*). Przy odczycie można spraw-  
dzić czy wystąpiło jakieś przerwanie CIA  
– no ale po co? Przecież jesteśmy za *SEI*  
i żadne przerwanie się nie uruchomiło?  
Tak naprawdę, rozkaz *SEI* nie wyłącza prze-  
rwań, tylko je maskuje – oznacza to tyle, że  
nawet jak wystąpią, to nie przerwą naszego  
programu, ale też nie zostaną one zaniecha-  
ne. Poczekaają, aż komenda *CLI* je odblokuje  
i wtedy od razu ruszą z kopyta. Istnieje więc  
duże ryzyko, że zaraz po naszym ustawieniu  
i po komendzie *CLI*, wywołają się wszystkie  
te które wystąpiły w międzyczasie. To nie jest  
dobra sytuacja, bo uruchamiają się jakby na  
„starych zasadach”, a z nowymi warunkami.  
Nam chodzi o wyłączenie niepotrzebnych,

a włączenie tylko tego, które chcemy i to jesz-  
cze nie od razu, ale w odpowiednim czasie!  
Dlatego zanim wyjdziemy z naszej konfiguracji  
– musimy skasować wszystkie znaczniki prze-  
rwań, aby później pierwsze które się wykona to  
właśnie nasze i to wtedy gdy tego oczekujemy.  
Przerwania CIA mamy załatwione, teraz usta-  
wiamy VIC i przerwanie rastrowe. Odpowie-  
dzialny za to jest rejestr *\$D01A*:

| \$D01A Interrupt Mask Register (IMR)<br>[def. \$00 (%00000000)] |  |
|---|--|
| Bit 7-4   | Always 1   |
| Bit 3   | Light-Pen Triggered IRQ Flag                         |
| Bit 2   | Sprite to Sprite Collision IRQ Flag (see \$D01E)     |
| Bit 1   | Sprite to Background Collision IRQ Flag (see \$D01F) |
| Bit 0   | Raster Compare IRQ Flag (see \$D012)                 |

Tu sprawa jest oczywista. Zaznacza-  
my *Bit0* odpowiedzialny za uruchomienie  
przerwania bazujących na osiągnięciu odpo-  
wiedniej linii rastra – w przeciwieństwie do  
rejestrów kontrolnych CIA, kasujemy przez  
gaszenie. A co z tym znacznikiem wywołane-  
go przerwania, dla przerwania VIC? Też jest –  
i tu szczególnie o tym powinniśmy pamiętać,  
bo zaraz po *\$D01A*, ustawiamy linię ekranu,  
w której nastąpi przerwanie (*\$D012*, *\$D011*),  
jeśli nie skasujemy znacznika wystąpienia  
przerwania, a pechowo trafimy w czas innego  
– tego, które działało przed naszym progra-  
mem - to wywołanie naszego przerwania, nie  
nastąpi w przez nas żądanej linii, ale zaraz po  
*CLI*, a dopiero następnym razem tam gdzie  
powinno. Tyle że tu mamy oddzielny rejestr  
do ustawiania oraz do sprawdzania i kasowa-  
nia – ten drugi to *\$D019*. W naszym przykła-  
dzie jest on umieszczony na samym końcu  
przed *CLI*, a to dlatego że ryzyko wystąpienia  
jakiegokolwiek innego przerwania po skaso-  
waniu będzie po prostu najmniejsze (ma na  
to tylko 2 cykle zegarowe – czyli czas trwania  
instrukcji *CLI*), potem... wywołanie nastąpi  
tam gdzie zaplanowaliśmy. Do rejestru *\$D019*  
wpisywana jest wartość *\$FF*, to dość ostrożna  
metoda, ale oznacza tylko tyle, że znacznik  
wykonania któregośkolwiek z przerwania VIC  
-a, będzie odnotowany.

| \$D019 Interrupt Request Register (IRR)<br>[def. \$00 (%00000000)] |  |
|--|--|
| Bit 7  | 1 = IRQ has been generated                           |
| Bit 3  | Light-Pen Triggered IRQ Flag                         |
| Bit 2  | Sprite to Sprite Collision IRQ Flag (see \$D01E)     |
| Bit 1  | Sprite to Background Collision IRQ Flag (see \$D01F) |
| Bit 0  | Raster Compare IRQ Flag (see \$D012)                 |

Co prawda, jak już wspominałem wcze-  
śniej, gdy komputer jest świeżo po resecie,  
to akurat przerwanie bazujące na VIC-u są  
wyłączone, czyli niebezpieczeństwo ich wy-  
stąpienia podczas naszego programu nam nie  
zagroza, ale nie zawsze mamy tak jasne warun-

ki, a często się też zdarza, że sami zmieniamy swoje jedne przerwania na drugie i o takich sprawach trzeba po prostu pamiętać – nie zawsze przecież inicjacja trwa krótko – często czyści się ekran i kolory, a dopiero potem następuje *CLI*. Jak nietrudno sobie wyobrazić, wtedy wystąpienie poprzedniego przerwania jest dużo wyższe, a czasem nawet i pewne. W listingu zostały jeszcze nie omówione rejestry *\$D012* i *\$D011*, ustawiające żadaną linię rastra, w której ma nastąpić przerwanie. Teoretycznie wystarczyłoby ustawić żadaną linię w samym *\$D012* – ekran C64 (PAL) ma 312 linii rastra, z czego widzialnych jest ok. 284, a ostatnia linia ekranu przed rozpoczęciem dolnej ramki to 249 – więc do wielu celów wystarczą wartości do 255, które reprezentuje właśnie ten rejestr.

| \$D012 Raster Position |   |
|------------------------|---|
| Bit 7-0                | Read : Get current Raster Position<br>Set Raster Position for IRQ |

| \$D019 Interrupt Request Register (IRR)<br>[def. \$00 (%00000000)] |  |
|--|--|
| Bit 7  | Raster Position Bit 8 from \$D012          |
| Bit 6  | Extended Color Text Mode: 1 = Enable       |
| Bit 5  | Bitmap Mode: 1 = Enable                    |
| Bit 4  | Blank Screen to Border Color: 0 = Blank    |
| Bit 3  | Select 24/25 Row Text Display: 1 = 25 Rows |
| Bit 2-0  | Smooth Scroll to Y Dot-Position (0-7)      |

Po przekroczeniu 256 linii, rejestr *\$D012* zaczyna zliczać rastry od zera, ale aby było jasne że to nie to samo zero co na samej górze ekranu to w komórce *\$D011* zapala się bit7 (najstarszy rastra). Przy ustalaniu linii, w której ma nastąpić przerwanie, on też uczestniczy! Czyli, jak ustawiamy np. linię 100 w *\$D012*, a zaniechamy ustawienia najstarszego bitu w *\$D011*, to czy cel zostanie ustawiony – zależy. Zależy od tego, w której linii akurat nastąpiło ustawienie – czy w górnych 256 czy poniżej. Jeśli w górnych – to się udało – przerwanie wystąpi w linii sto, bo w tym czasie najstarszy bit *\$D011* był wyzerowany. Podobnie jak *\$DC0D*, rejestry te mają podwójną „naturę”, czyli inaczej reagują na zapis i odczyt. Oznacza to również, że jeśli ustawienie *\$D012* nastąpiło w chwili, gdy raster przekroczył 256 linię ekranu, to żądana linia rastra nie będzie w 100, tylko 256+100 czyli 356! Ta linia nie istnieje i nasze przerwanie nigdy nie zostanie wykonane. Ponieważ nie wiemy, kiedy dokładnie następuje ustawienie rejestru rastra w naszym programie, więc ustawienie siódmego bitu w rejestrze *\$D011* jest wręcz niezbędne („dla spokojności sumienia”). Przerwanie zostało ustawione, teraz wypadaloby je uruchomić i sprawdzić efekty.

Kolor tła zmienia się w stałym miejscu

ekranu. Ok, ale jak przyjrzyć się bliżej, to tam na początku dzieje się niezła „szarpanina”, uuu... czemu? Jest tu szereg czynników, które mają na to wpływ. Należy się zastanowić, co tak naprawdę dzieje się w chwili, w której ma nastąpić przerwanie. Jak już wspomniałem wcześniej, kiedy pojawi się sygnał przerwania, to procesor nie urywa aktualnie wykonywanej instrukcji programu „w połowie”, a czeka aż się wykona i dopiero wtedy może się zająć rozpoczęciem procesu przerwania. Mało tego – przerwanie zanim w ogóle zacznie swój proces, ma jeszcze 2 cykle opóźnienia, a dopiero potem bierze się do działania. Czyli, jak osiągniemy tą naszą przykładową linię 100, to najpierw miną 2 cykle - a zwykła linia rastra ma ich 63, potem dokończona zostanie instrukcja aktualnie wykonywana – tu już pojawia się rozbieżność, bo może to być 0 (idealne trafienie), albo nawet i 6 – bo niektóre instrukcje mogą zajmować nawet 7 cykli (a niepublikowane nawet 8), czyli jak trafimy na pierwszy cykl takiej 7-demki – to musimy poczekać pozostałe 6. Następnie zostaje wykonana instrukcja wejścia w przerwanie IC (podobna jest do komendy *BRK*),

| Interrupt command 'IC' |          |     |  |
|------------------------|----------|-----|--|
| #                      | address  | R/W | Description  |
| 1                      | PC       | R   | fetch opcode, increment PC                                   |
| 2                      | PC       | R   | read next instruction byte (and throw it away), increment PC |
| 3                      | \$0100,S | W   | push PCH on stack (with B flag set), decrement S             |
| 4                      | \$0100,S | W   | push PCL on stack, decrement S                               |
| 5                      | \$0100,S | W   | push P on stack, decrement S                                 |
| 6                      | \$FFFE   | R   | fetch PCL  |
| 7                      | \$FFFF   | R   | fetch PCL  |

która trwa kolejne 7 cykli. W pierwszym cyklu procesor pobiera rozkaz (IC), w drugim czyta bajt skojarzony z rozkazem (czyli następny bajt) – i tu należy się małe wyjaśnienie. Zasada działania procesorów serii 6500, wymusza zawsze przynajmniej dwa odczyty dla każdej instrukcji, jest to bardzo efektywne w ogólnym rozrachunku, ale w przypadku jednobajtowych mniej, – dlatego też np. instrukcja *NOP* i *LDA #\$00*, zajmują tyle samo czasu procesora. Co się dzieje z drugą daną po instrukcji jednobajtowej – po prostu jest odrzucana, nie inaczej jest w tym przypadku. W trzecim i czwartym cyklu zapisywany jest na stosie (czyli takim buforze danych dla procesora) adres programu w którym go przerywamy – w końcu po wykonaniu przerwania trzeba do niego wrócić, nie? W piątym cyklu zapisujemy rejestr znaczników – bo jak przerywaliśmy np. jakieś dodawanie z przeniesieniem, to po powrocie może się okazać że wynik będzie nieprawidłowy – słowem, zapisujemy wszystko tak, aby po powrocie odtworzyć dokładnie te same warunki. Szó-

sty i siódmy cykl to pobranie adresu pamięci, w którym znajduje się program przerwania, no i zaraz potem się tam przenosimy.

Ile minęło? Przy pomyślnych wiatrach 2+0+7 = 9 cykli, przy mniej nawet 2+6+7 = 15. Jak widać na rozpisce instrukcji IC, adres do programu przerwania znajduje się w komórkach *\$FFFE/\$FFFF* (ostatnie dwa bajty pamięci kommodorka) – zaraz, zaraz a gdzie nasze *\$0314/\$0315*? No właśnie, już wspominałem przy pierwszym z listingów, że tak naprawdę to doklejamy się do systemu – i to jest prawda. Adres wskazany w *\$FFFE/\$FFFF* jest standardowo - czyli po uruchomieniu C64 - pobierany z ROM-u. A co to jest za adres, sprawdźmy:

| IRQ Entry      |              |          |
|----------------|--------------|----------|
| FF48: 48       | PHA          | ;3       |
| FF49: 8A       | TXA          | ;2       |
| FF4A: 48       | PHA          | ;3       |
| FF4B: 98       | TYA          | ;2       |
| FF4C: 48       | PHA          | ;3       |
| FF4D: BA       | TSX          | ;2       |
| FF4E: BD 04 01 | LDA \$0104,X | ;4       |
| FF51: 29 10    | AND #\$10    | ;2       |
| FF53: F0 03    | BEQ \$FF58   | ;3 lub 2 |
| FF55: 6C 16 03 | JMP (\$0316) | ;5       |
| FF58: 6C 14 03 | JMP (\$0314) | ;5       |

O, pojawiło się nasze *\$0314*, a przed nim... jeszcze cały ciąg instrukcji. Zrzut rejestrów X, Y i Akumulatora na stos – słusznie, program w przerwaniu na pewno będzie je zmieniał, a po przerwaniu powinny tam być dokładnie te same wartości. Potem sprawdzenie rejestru znaczników (wrzuconego wcześniej na stos przez instrukcję IC), czy przypadkiem przerwanie nie było spowodowane instrukcją *BRK* – jeśli nie, to skaczymy pod adres wskazany w *\$0314/\$0315*, a tam w końcu do naszego programu mrugania tłem. Uff, ile mogło czasu upłynąć – dodajemy (przydatna umiejętność przy cyklowaniu) – obok listingu wypisałem przy każdej instrukcji ilość cykli potrzebną do jej wykonania – w sumie od *\$FF48* do naszego *DEC \$D021*, upłynie 29 cykli. Dodając jeszcze cykle od początku rozpoczęcia przerwania daje nam to wartość w granicach 38..44! To prawie cała linia rastra – wizualnie dotarcie do naszego programu od początku linii nastąpi w jakieś 26-32 kolumnie ekranu znakowego (1 cykl = jedna kolumna) – i właśnie w tych granicach nasz program charakteryzuje się niekończącą się szarpaniną.

Czy nie dałoby się tego jakoś przyspieszyć, można, ale na razie skupmy się na ustabilizowaniu tego co jest. Na pierwszy ogień pójdzie często wykorzystywana procedura w starych kursach C&A [Pic. 2]:

```

irq      ldy #$2e    ;cyklowanie
        cpy $d012
        bne *-3
        ldx #$09
        dex
        bne *-1
        nop
        nop
        nop
        iny
        cpy $d012
        beq ++4
        nop
        nop
        ldx #$09
        dex
        bne *-1
        nop
        nop
        iny
        cpy $d012
        beq ++4
        bit $ea
        ldx #$0a
        dex
        bne *-1
        nop
        iny
        cpy $d012
        bne ++2

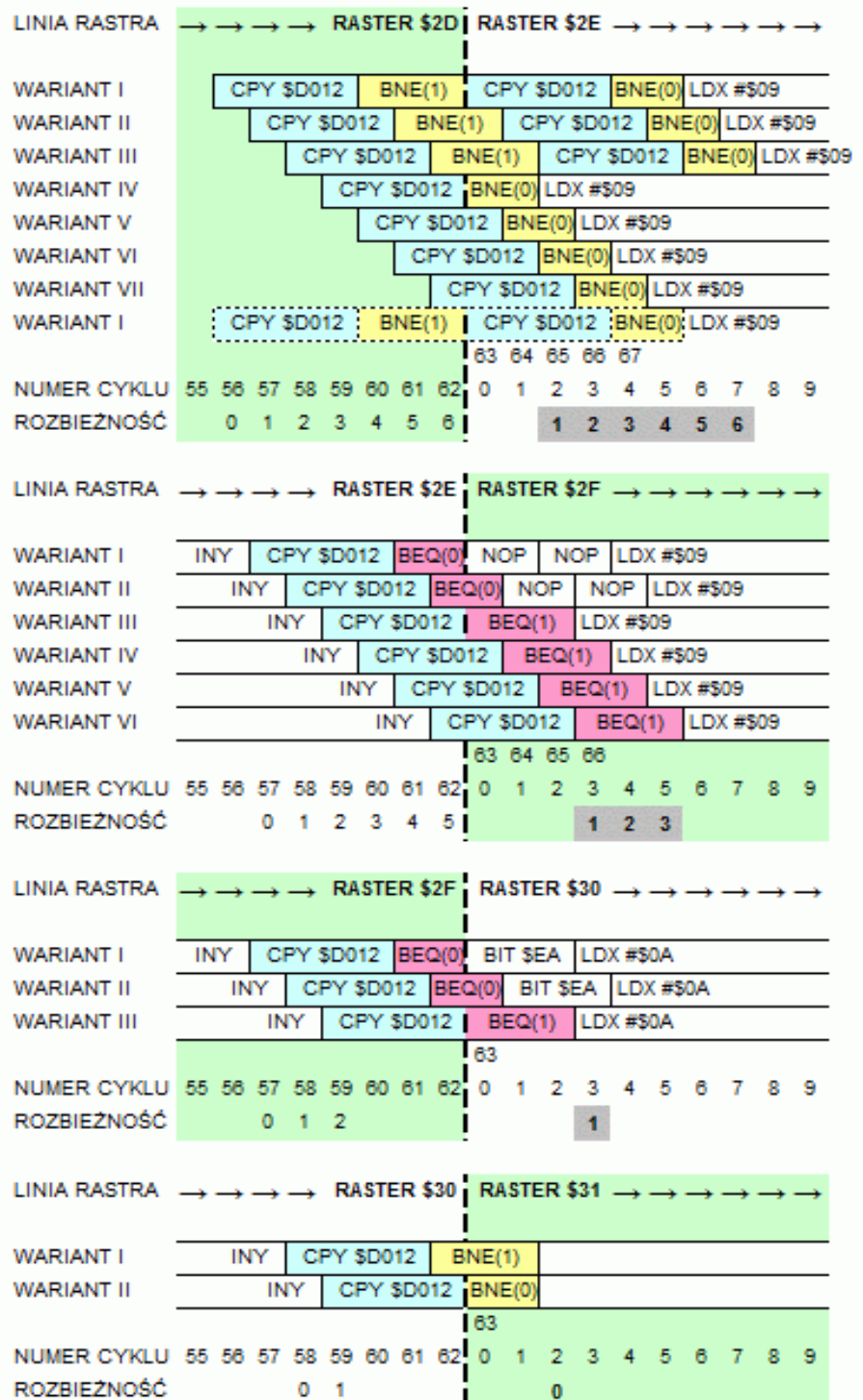
```

Pic.2

Wstrzymajmy się jednak, na razie z umi-  
-szczeniem tego kawałka kodu w naszym  
przerwaniu. Owszem jest to jedno z działają-  
cych rozwiązań, ale nie jest pozbawione wielu  
wad. Po pierwsze, jest ono do wykorzystania  
w dość konkretnym miejscu, tu musi się za-  
czynać od linii rastra \$2D, po drugie potrze-  
buje poświęcić aż 5 linii rastra na wykonanie  
swojego zadania. Po trzecie w listingu znajdu-  
je się mnóstwo części krytycznych – to znaczy  
zależnych od fizycznego umiejscowienia ich  
w pamięci – mowa tu o pętlach, które mogą się  
znaleźć na skraju przecięcia się 256 bajtowych  
bloków pamięci – a wtedy pewne instrukcje  
zabierają dodatkowe cykle procesorowi i cała  
synchronizacja może okazać się nieskutecz-  
na. Procedura jest również wrażliwa na różne  
stany pracy VIC-a (*Badlines*, *Sprites*). Ostat-  
nią niekorzystną cechą jest też rozmiar proce-  
dury – jak widać do najkrótszych nie należy,  
a co jeśli potrzeba nam kilka takich przerw-  
wań? Spróbujmy ją jednak, chociaż przeanaliz-  
ować i sprawdzić jak powinna działać. Cała rzecz  
opiera się na założeniu, że każda z linii rastra  
oferuje do wykorzystania

63 cykle, i opierając się o tę wartość oscylujemy  
opóźnieniami w kolejnych liniach, eliminując  
rozbieżności, aby na końcu uzyskać niepewność  
równą zero – skomplikowane? Zaraz się okaże.  
Pierwszą linię – tą, na którą było ustawione prze-  
rwanie (\$2D) – sobie odpuszczamy od cyklowa-  
nia – jak wyliczałem wcześniej, wejście do nasze-  
go przerwania następuje dość późno i najbardziej  
jest narażone na losowe wypadki, dlatego też na  
wstępie procedury od razu jest oczekiwanie na  
kolejną linię, czyli \$2E. Gdy to już nastąpi nasza  
rozbieżność w cyklach zmniejsza się do zakresu  
0..5 cykli. Na obrazku Pic.3, u samej góry, widać  
wszystkie możliwe warianty przejścia z linii ra-  
stra \$2D do \$2E i to co otrzymujemy potem. W  
najgorszym przypadku porównanie *CPY \$D012*

wypadnie cykl przed zmianą linii, wtedy po *BNE*  
jeszcze raz nastąpi porównanie z *\$D012* i dopiero  
przejście do dalszej części kodu. Później mamy  
pętlę na 9 powtórzeń *DEX*, która łącznie zabiera  
46 cykli, trzy *NOP*-y (6 cykli), jeden *INY* (2), i ko-  
lejne porównanie *\$d012* (4). Linię \$2E zaczęliśmy  
od 2-8 cykła, opóźnienie pozostałych instrukcji to  
58 – daje to po kolejnym *CPY \$D012* cykl z prze-  
działu 61 do 66, czyli tylko w dwóch przypadkach  
mieścimy się jeszcze w „starej” linii (cykle 61,62),  
i właśnie te korygujemy poprzez *BEQ* – wszystkie  
pozostałe omijają *NOP*-y opóźniające. Rozbież-  
ność maleje nam o 3 cykle. W kolejnej linii, w po-  
dobny sposób pozbywamy się kolejnych dwóch  
cykli, a w następnej walczymy tylko z jednym cy-  
klem, na koniec uzyskując pełną synchronizację.



Pic.3

Wyrównywanie cykli, to problem zależny od programu działającego poza przerwaniem, bo to w nim właśnie są nieznanne instrukcje, które powodują początkowe opóźnienie przerwania. A gdyby tak, używać tylko określonego typu instrukcji, np. maksymalnie 4 cyklowych? Bzdura, nie chodzi nam przecież o ograniczanie w jakimkolwiek stopniu programów. Jednak w końcu ktoś wpadł na to, że tuż przed momentem, w którym ma się wykonać przerwanie, można stworzyć mu bardzo przyjazne warunki. Pytacie się jak? – poprzez drugie przerwanie.

Technicznie wygląda to w ten sposób, że ustawiamy przerwanie też trochę wcześniej niż docelowa linia, ale teraz wystarczy „zmarnować” tylko dwie, aby uzyskać pełną synchronizację. A więc mamy pierwsze przerwanie – wprowadzające, a jego zadaniem jest jak najszybsze ustawienie drugiego, tego właściwego (o linię dalej) i rozpoczęcie niekończącego się ciągu instrukcji NOP – który dla przerwania nr dwa zasymuluje zwykły program „nie w przerwaniu”:

```

IRQ
inc $d012 ;6 (+ MIN.9)cykli
stx SAVE_X+1 ;4
ldx #<irq_2 ;2
stx $fffe ;4
ldx #>irq_2 ;2
stx $ffff ;4
INC $d019 ;6
tsx ;2
cli ;2
nop ;2
nop ;2
... ;i jeszcze Z 10 razy NOP

irq_2
txs ;2 (+ 9 albo 10)
INC $d019 ;6
PHA ;3
sty SAVE_Y+1 ;4
lda #<irq ;2
ldx #>irq ;2
ldy #xx ;2
sta $fffe ;4
stx $ffff ;4
sty $d012 ;4
inc $d019 ;6
CMP $00,X ;4
CMP $00 ;3

lda $d012 ;4
cmp $d012 ;4
beq *+2 ;2 lub 3

DEC $d021
...

IRQ_EXIT

```

```

SAVE_Y ldy #$00
SAVE_X ldx #$00
PLA
rti

```

No więc co my tu mamy – na początku trafiamy do *IRQ\_1*, a pierwsza instrukcja to *INC \$D012*, która powoduje że określamy nową linię rastra do wywołania – którą? – aktualną plus jeden, czyli następną linię. Potem jest *STX ...*, no właśnie – tu zaczynają się nowości, *STX* jest, ale *LDX* nie ma. Jak popatrzmy na kolejne instrukcje, to zauważymy wspomniane wcześniej nasze *\$FFFE/\$FFFF*, czyli adres przerwania zawarty w ROM-ie? Tak, ale teraz nie w ROM-e tylko w RAM-ie. Otóż, nie musimy koniecznie korzystać z procedur systemu zawartego w Kernalu, i można go wyłączyć. W naszym poprzednim programie zamieniamy linie z *\$0314/\$0315* i dodajemy wyłączenie ROMu (Kernal, a Basic też nam jest zbędny) za pomocą komórki *\$01*:

```

SEI
LDA #$35
STA $01
LDA #<IRQ
STA $fffe
LDA #>IRQ
STA $ffff
...

```

Tylko co wtedy się stanie z wejściem do przerwania (*\$FF48*)? Napiszemy sobie własne – nawet zoptymalizowane, przecież w naszym programie nie ma potrzeby sprawdzania czy wystąpiła np. instrukcja *BRK*. Nawet nie musimy zapamiętywać stanów wszystkich rejestrów – jeśli nie będziemy z nich korzystać w programie przerwania. Tak więc teraz, gdy nastąpi przerwanie, i wykona rozkaz *IC*, to trafi od razu do naszego programu – oznacza to, że przy normalnych warunkach czas od wzbudzenia przerwania do wejścia do naszej procedury wyniesie ok.  $2+7+(0.6) = 9.15$  cykli. No, oczywiście trzeba pamiętać o tym, że teraz my się musimy zatroszczyć o zachowanie stanu rejestrów X,Y i A sprzed przerwania i przy wyjściu je odtworzyć. Wracamy do programu, a więc ustaliliśmy nową linię rastra dla przerwania i również zmieniamy adres *\$FFFE/\$FFFF* na *IRQ\_2*, potem zwiększamy rejestr *\$D019* – to taki trik oszczędzający pamięć, zamiast wpisywania *LDX #\$01, STX \$D019* – czyli informujemy komputer, że przerwanie wykonało się. Dalej jest instrukcja *TSX* – czyli prześlij wskaźnik stosu do rejestru X, a to po to, żeby jak wywołamy przerwanie numer 2, a potem je opuścimy to nie było potrzeby wracać do tego

pierwszego, ale od razu do kodu który przerwaliliśmy przerwaniem nr 1. Zapisując indeks, a w *IRQ\_2* go odtwarzając, zamazujemy informację o podwójnym wejściu, która odkłada się na stosie (patrz instrukcja *IC*) – po wszystkich wracamy bezpośrednio do zwykłego programu. Po *TSX*, jest *CLI*, a co to?

No właśnie, gdy następuje przerwanie, znacznik maski przerw w rejestrze znaczników także ustawia się na 1 – czyli wykonuje takie *SEI*, żeby inne przerwania mu nie przeszkadzały. Wykonując *CLI*, traci status ochronny i jest traktowany jako zwykły program nie w przerwaniu. To dziwna sytuacja, bo przerwanie niby się skończyło, ale nie wraca do programu który przerwało - brakuje instrukcji *RTI*. Właśnie tu dochodzimy do kwestii wychodzenia z przerwania, temat ten do tej pory nie był poruszany.

W pierwszym listingu wyjście z przerwania to instrukcja *JMP \$EA31*, czyli gdzieś do Kernala. Ten adres jest również standardowo umieszczany w komórkach *\$0314/\$0315* po resecie, my się właśnie wklejaliśmy między *\$0314/\$0315*, a te tajemnicze *\$EA31*. Co tam jest? No obsługa klawiatury, kursora i innych rzeczy systemowych – które są niezbędne do korzystania z C64 zaraz po resecie. Nas interesuje w tym momencie tylko część odpowiedzialna za opuszczenie przerwania. Oto fragment systemu:

```

EA31:
...
IRQ_EXIT
EA7B: 20 87 EA JSR $EA87
; Scan Keyboard
EA7E: AD 0D DC LDA $DC0D
; CIA1: Interrupt (IRQ)
Control Register
EA81: 68 PLA
EA82: A8 TAY
EA83: 68 PLA
EA84: AA TAX
EA85: 68 PLA
EA86: 40 RTI

```

W *\$EA7B* jest jeszcze skok do sprawdzenia klawiatury, a potem szereg instrukcji zakończonych rozkazem *RTI* – czyli wyjściem z przerwania, a ta instrukcja to taka odwrotność rozkazu *IC*. *LDA \$DC0D*, spełnia tu rolę taką samą, jak przy naszym programie inicjacji – powiadamia, że przerwanie *CIA* się odbyło, potem jest rzut ze stosu, zachowanych przy wejściu (*\$FF48*) rejestrów, następnie *RTI*, i to tyle – po przerwaniu. Jak widać można już sobie skrócić *JMP \$EA31* na *\$EA7B* (uwzględniamy obsługę klawiszy), albo *\$EA7E*, albo nawet *\$EA81* – gdy nie korzystamy z przerwania *CIA*. W przypadku na-

szego ostatniego programu, nic sobie nie możemy skrócić, bo ... wyłączyliśmy już ROM Kernala, a więc instrukcję wyjścia musimy uzupełnić sobie sami.

| RTI command |          |     |  |
|-------------|----------|-----|--|
| #           | address  | R/W | Description                                    |
| 1           | PC       | R   | fetch opcode, increment PC                     |
| 2           | PC       | R   | read next instruction byte (and throw it away) |
| 3           | \$0100,S | R   | increment S                                    |
| 4           | \$0100,S | R   | pull P from stack, increment S                 |
| 5           | \$0100,S | R   | pull PCL from stack, increment S               |
| 6           | \$0100,S | R   | pull PCH from stack, increment S               |

Wracając do ostatniego listingu, po *CLI* są już tylko *NOP*-y, czyli zostajemy w takim bezruchu - oczekiwaniu. Na co? Oczywiście na wywołanie przerwania *IRQ\_2*, nic nie ruszamy, *NOP*-ów powinniśmy nastawiać tyle, aby starczyło do czasu osiągnięcia kolejnej linii rastra, w której ma nastąpić przerwanie. Co nam to daje, ano to że w końcu *IRQ\_2* zostanie wzbudzone, ale tym razem znamy dokładnie jego warunki wywołania, tu już nie ma przypadkowych instrukcji które zostają przerywane - w *IRQ\_2* wchodzimy tylko z 1 cyklem niepewności, bo albo trafiło idealnie po którymś *NOP*-ie, albo w połowie *NOP*-a (+1 cykl). Co więcej, dane w rejestrach też się nie zmieniły - oprócz *X* - który sobie zachowaliśmy w *IRQ\_1*, a w międzyczasie służył także do przechowania wskaźnika stosu. Teraz wystarczy nam zredukować tylko ten jeden cykl, poświęcając pierwszą linię *IRQ\_2*, przy okazji przestawiamy wektor przerwania na *IRQ\_1*, bo w kolejnym wywołaniu trzeba zacząć wszystko od nowa, odtwarzamy teraz również wskaźnik stosu (*TXS*), zachowujemy akumulator i rejestr *Y* (wcześniej nie było potrzeby), jeszcze raz potwierdzamy wykonanie przerwania w *\$D019* i po *BEQ \*+2*, mamy cel osiągnięty. Instrukcje *CMP \$00*, pełnią tu rolę czysto opóźniającą.

Wyjście z *IRQ\_2* nie potrzebuje chyba dodatkowego komentarza - odtworzenie rejestrów i *RTI*.

Metoda ta jest dość często wykorzystywana, całkiem niezawodna, oraz likwiduje większość wad poprzedniego rozwiązania. Jeśli zadbamy o to, by etykiety *IRQ\_1* i *IRQ\_2* były w tym samym bloku/stronie pamięci c64, to nawet nie trzeba korygować starszego bajtu wektora przerwania (*\$FFFF*).

Przy temacie cyklowania, grzechem byłoby nie wspomnieć o różnych pułapkach. Bo wszystko to fajnie działa, dopóki mamy do czynienia z „czystymi” liniami rastra. Winowajców jest zazwyczaj dwóch - *Badlines* i *Sprites* - bezpardonowo zabierają cykle procesorowi, i nie pozostaje nic innego jak dostosowanie się do ich warunków. Temat jest

w zasadzie znany - wspominałem go w art. o *NUFLI*, ale tak dla przypomnienia...

W każdej pierwszej linii wiersza znakowego, układ graficzny *VIC* pobiera z pamięci wskaźniki do znaków, które wyświetla przez 7 kolejnych linii. To pobranie zajmuje dokładnie 40 cykli, czyli dokładnie 1 cykl na 1 znak. Wchodząc głębiej w tą kwestię, trzeba nadmienić, że na każdy cykl, przypada część procesorowa i *VIC*-a (po pół) - *VIC* cały czas wyświetla, *CPU* liczy. Niestety w tych sytuacjach połówka *CPU* jest po prostu zabierana, *VIC* potrzebuje więcej czasu na swoje zadania, a jak widać on tu rządzi.

Podobnie jest przy dźwiękach, każdy z nich zabiera po dwa cykle w linii rastra, w której jest wyświetlany. Niestety to nie wszystko, proces blokowania *CPU*, poprzedzają jeszcze 3 cykle na „dokończenie” ostatniej instrukcji, ale pod warunkiem, że jest to jej część korzystająca z zapisu - przy tabelkach instrukcji (np. *IC*) kolumna *R/W* - w innych wypadkach te 3 cykle to zwykła strata. Można trochę uszczęknąć z tych 3 cykli ustawiając niektóre instrukcje (*RMW*) tak, aby ich część odpowiedzialna za zapis znalazła się właśnie w tej strefie. Każdy z 8-miu *Sprites* dostępnych na *C64* ma również określone miejsce, w którym kradnie te cykle (patrz *Pic.5*). Trzeba po prostu o tych sprawach pamiętać i umieć korygować te cykle przy cyklowaniu. Najlepiej unikać wywołań przerw w *Badlines*. Przy pisaniu kodu opierającego się o wyliczanie cykli, należy też zwrócić uwagę na granice stron pamięci, gdyż w wielu instrukcjach to zdarzenie dodaje po jednym cyklu dodatkowo. W kompilatorze *64tass* jest do tego dyrektywa *.page-.pend*. Umieszczając w niej takie krytycznie części kodu, przy kompilacji zostanie wypisane ostrzeżenie, jeśli taka sytuacja nastąpi - wtedy np. wystarczy trochę zmodyfikować kolejność niektórych instrukcji. Należy też pamiętać o „zabawach” z komórką *\$01*, przerwanie rastra bazuje na porównywaniu rejestru *\$D012*, a więc jak go „wyłączymy” mogą nastąpić nieoczekiwane zdarzenia. Na początku artykułu była też wzmianka o przerwaniach *CIA*, a w zasadzie do tej pory je tylko zapobiegawczo wyłączaliśmy - może warto z nich również korzystać? Czemu nie, samo wywoływanie wygląda tak samo jak przy *IC*, różni się oczywiście zdarzeniem, po którym to następuje. Ustawiamy sobie jakąś wartość początkową - jednostką jest tu cykl - i komputer po odliczeniu do zera wywołuje przerwanie - proste.

Takich liczników możemy sobie ustawić nawet cztery, po dwa na każdy *CIA* - wartość początkowa to liczba z przedziału 0-65535 - czyli o długości słowa 16-bit. Jeśli dodamy do tego, że na jedno odświeżenie ekranu komputer potrzebuje (63\*312) 19656 cykli, a odświeża ekran („ramkę”) 50 razy na

sekundę to maksymalnie wychodzi ok. 3,33 „ramki”, czyli jakieś 7 ms. - dane te odnoszą się do systemu *PAL*. Może to i niedużo, ale dla przerw rastrowych i tak nie osiągalne. Drugą cechą tych przerw jest wybór ich powtarzalności. Przy *\$D012* - było jasne, że mowa jest o częstotliwości raz na ramkę i to w sposób nieskończony. Tu można ustawić ciągłość lub jeden „strzał” - w dodatku wywołanie może dotyczyć dowolnego miejsca, a nie tylko początku linii rastra. Trzecia cecha, to nowość - jeśli korzystamy z drugiego układu *CIA* to przerwania mają priorytet *NMI* i są nie maskowalne dla instrukcji *SEI*! Co to wszystko oznacza - ano tyle, że przy konfiguracji przerw mamy zupełną swobodę i możemy sobie praktycznie dowolnie mieszać wywoływanie tych przerw, a nawet stworzyć ich dwa poziomy - znaczyłoby to, że podczas wykonywania jednego z przerw można je zatrzymać i wywołać przerwanie drugiego poziomu (!)

Tyle teorii, spróbujmy stworzyć jakieś przerwanie „timerowe”:

```
sei
lda #$35
sta $01

ldx #$00
stx $d01a
dex
stx $d019

lda <CIA2_IRQ
sta $ffa
lda >CIA2_IRQ
sta $ffb

lda <CIA1_IRQ
sta $ffe
lda >CIA1_IRQ
sta $fff

lda <400      ;interval 1 (cycles)
ldx >400
sta $dd04
stx $dd05

lda <450      ;interval 2
ldx >450
sta $dc04
stx $dc05

lda #%10010001
sta $dd0e
sta $dc0e

lda #%10000011
sta $dc0d
bit $dc0d
bit $dd0d
```

```

sta $dd0d
cli
jmp *

CIA2_IRQ
bit $dd0d
dec $d020
rti

CIA1_IRQ
bit $dc0d
inc $d020
rti

```

W zasadzie jak się przeczyta opisy użytych rejestrów CIA, to komentarz jest zbędny. Nie ma tu żadnych sztuczek i pułapek, a tylko ustawianie to czego żądamy. Zasadnicza różnica jest w adresie wektora wywołania – układ CIA2 korzysta z adresu \$FFFA/\$FFFB (NMI), a układ CIA1, tak jak przy przerwaniach rastrowych, z \$FFFE/\$FFFF. Poszczególne rejestry ustawiające, mają swoje lustrzane odbicie (\$DCxx-\$DDxx), i również ich sposób działania jest identyczny, różnica polega na maskowalności przerwania. Tu wygrywa CIA2 i w związku z tym ma również wyższy priorytet działania. Oba układy mają po dwa liczniki 16-to bitowe, i można używać ich równocześnie i niezależnie, jednakże należy zwrócić uwagę na to, że para liczników przypadająca na każdy układ odnosi się do tego samego wektora wywołania. Musimy więc sami zatroszczyć o rozróżnienie który z pary liczników spowodował przerwanie – sprawdzenie to, realizuje się za pomocą rejestrów kontrolnych przerwań \$Dx0D, w których dwa najmłodsze bity informują właśnie, który licznik wywołał przerwanie. Przykładowo może to wyglądać tak:

```

CIA1_IRQ
pha
lda $dc0d
lsr
bcc timer_B

TIMER_A
dec $d020
pla
rti

timer_B
DEC $d021
pla
rti

```

Gotujemy tu dwie pieczenie na jednym ogniu, sam odczyt z \$DC0D – jak informowałem wcześniej – czyści znacznik wystąpienia

przerwania, czyli i tak byłby powinnością, aby można było „zakończyć” przerwanie. Wracając do listingu setup-u CIA, ustawione są tam dwa liczniki, z dwóch różnych układów – jak widać, wartości wpisywane do poszczególnych rejestrów „pokrywają” się dla obu. Różnica jest tylko w liczbie cykli do odliczania (interval). Oprócz rejestrów wartości liczników \$Dx04/\$Dx05, rejestrów kontrolnych przerwań CIA \$Dx0D, ustawiane są jeszcze rejestry kontrolne timer-a A \$Dx0E, które ustalają ciągłość odliczania (wywoływania), i tryb (PAL - 50Hz).

#### \$DD0E Control Register A [def. \$08 (%00001000)]

|       |  |
|-------|--|
| Bit 7 | Time-of-Day Clock Frequency: 1 = 50 Hz, 0 = 60 Hz    |
| Bit 6 | Serial Port (SDDOC) I/O Mode: 1 = Output, 0 = Input  |
| Bit 5 | Timer A Counts: 1 = CNT Signals, 0 = System 02 Clock |
| Bit 4 | Force Load Timer A: 1 = Yes                          |
| Bit 3 | Timer A Run Mode: 1 = One-Shot, 0 = Continuous       |
| Bit 2 | Timer A Output Mode to PB6: 1 = Toggle, 0 = Pulse    |
| Bit 1 | Timer A Output on PB6: 1 = Yes, 0 = No               |
| Bit 0 | Start/Stop Timer A: 1 = Start, 0 = Stop              |

Pozostałe rejestry w tych układach odnoszą się już do ustawiania transmisji z urządzeniami zewnętrznymi, a więc wykraczają poza temat. Kod w zasadzie nie robi nic ciekawego – znów zmienia kolor, tym razem ramki – istotna jest tu jednak zasada działania między NMI a IRQ. Wystarczy zlikwidować CLI, i okaże się że tylko jedno z przerwań będzie wywoływane. Dlatego też kluczowe ustawienia CIA2, czyli przerwań niemaskowalnych wykonywane są na samym końcu procedury inicjacji, aby zminimalizować ryzyko przerwania kodu w trakcie – w końcu te przerwania są odporne na SEI. Listingi jak i programy są również dostępne w pliku na naszej stronie internetowej. [Pic.4]

A co z cyklowaniem za pomocą przerwań timerowych? No teoretycznie powinno być tu dużo łatwiej w końcu ustawiamy je, już z dokładnością do jednego cykła. Niestety samo wywoływanie przerwania, rządzi się tymi samymi prawami co przy „rastrze”, a więc zanim dojdzie do naszej procedury musi dokończyć aktualną instrukcję wykonywaną przez procesor, potem rozkaz IC i... zostajemy z tym samym problemem jak poprzednio, a nawet gorzej – przy niespodziewanych opóźnieniach (Badlines, Sprites) nie znamy miejsca aktualnego odniesienia, jak to było przy porównywaniu z \$D012. Pewnie, dlatego te przerwania nie są tak popularne jak rastrowe. Należy jeszcze wyjaśnić zasadę działania licznika. Odlicza on od wartości przez nas ustawionej do zera... a właśnie że nie, bo do 1, potem – jeśli było ustawienie cykliczne, wstawia ponownie wartość startową. Ale wartość startowa trwa w nim przez dwa cykle, czyli jak ustawimy np. 8, to kolejne fazy będą wyglądać tak: 8, 8, 7, 6, 5, 4, 3, 2, 1, 8, 8, 7... Troszkę to komplikuje sprawę z ustalaniem aktualnego cyklu, ale o tym nie teraz.

Niewątpliwą zaletą przerwań CIA, są ich dwa rodzaje, z których jedno mają priorytet nad pozostałymi (NMI), i pozwalają na stworzenie dwóch warstw wywoływania nad „zwykłym” kodem. Oczywiście można je również mieszać z przerwaniami rastrowymi.

Ale czy można zrobić dwupoziomowe przerwanie rastra? Można, ale z pewnymi ograniczeniami. Jak? Już nawet takie zrobiliśmy przy cyklowaniu przez podwójne przerwanie. Różnica objawia się, za czyszczeniem flagi przerwania CLI, wtedy też nie ruszamy wskaźnika stosu, tylko spokojnie dalej wykonujemy kod (oczywiście nie chodzi tu o same NOP-y), i zmieniamy przerwanie w „zwykły” kod, który może być w dowol-



Pic.4

nym momencie przerwany przez przerwanie numer 2. Co się wtedy dzieje? W zasadzie wszystko idzie tak jak powinno tyle, że po zakończeniu tego „przerwania w przerwaniu” po rozkazie *RTI* wracamy do naszego pierwszego przerwania, a dopiero po kolejnym *RTI* wracamy do „zwykłego” kodu. Czy to ma sens? Czemu nie – wyobraźmy sobie sytuację gdy chcemy wyświetlić kilka rasterbarów, wklepujemy nasz docyklowujący kod, i jest ok, ale chcemy jeszcze aby grała muzyeczka w każdej ramce – można zrobić oczywiście dla niej oddzielne przerwanie, ale zakładamy wariant, że jeszcze dla atrakcyjności prezentacji korzystamy ze Sprites, a ponieważ z tych Sprites tworzymy duży ruchomy obiekt to jeszcze je powielamy warstwami, które wymagają dodatkowych wywołań przerwania. Używając kilku oddzielnych przerwania pierwszego poziomu, oprogramowanie takiej sytuacji jest oczywiście możliwe, ale trochę się trzeba przy tym namęczyć i pilnować, aby jedno przerwanie „nie weszło” na drugie, bo czeka nas wtedy sromotna porażka. Drugim rozwiązaniem jest użycie przerwania licznikowych, tu już mamy większy komfort jeśli chodzi o „zachodzenie” się przerwania, a jak jeszcze użyjemy niemaskowalnych to już w ogóle, z tym że tu trzeba się trochę pobawić przy samym wyliczaniu wartości dla licznika, aby trafił z wywołaniem w to miejsce, które chcemy. Trzecie rozwiązanie to właśnie ostatnie rozważanie z dwoma przerwaniem rastra – dla założonych warunków – ustawiamy najpierw przerwanie rastra na rasterbary, wywołujemy przerwanie numer 1, docyklowanie, zmieniamy kolory rastra, ustawiamy przerwanie nr 2 dla Sprites, potem *CLI*... a potem normalnie skaczemy do playera muzyczki, i nawet jeśli zaraz wywoła się przerwanie dla duszków to i tak po powrocie wróci do playera muzyczki kontynuując robotę - jedno na drugie już nie „najedzie”. [Pic.5]

### Jak „robią to” najlepsi?

Na koniec nie pozostało nic innego, jak pokazać połączenie wykorzystania w precyzyjnym cyklowaniu przerwania rastrowych i licznikowych. A warto je poznać, bo jak się okazuje, właśnie ten sposób jest wykorzystywany obecnie przez czołowe grupy scenowe. Właściwie to zdradzę że ostatni listing ukazujący tą metodę skutecznego przerwania, wzorowałem na procedurach znalezionych w ostatnim demie grupy *Booze Design & Instinct* o nazwie „*Mekanix*” – a więc nie z byle czego ;-)

Metoda ta łączy w sobie wszystkie pozytywne cechy wyżej wymienionych i jest też skuteczna. Przy poprawnej konfiguracji początkowej, daje możliwość wycyklowania praktycznie już w pierwszej linii wywołania rastra! Czyli nie dość, że szybka, zajmuje mało miejsca w pamięci, to jeszcze takie przerwanie można wywoływać bardzo często - zadowolić to może chyba najbardziej wybrednych i daje pole do popisu dla niesamowitych efektów. Jak działa? Potrzebną teorię już mamy opisaną wcześniej.

Kluczem jest tu odpowiednie uruchomienie jednego z liczników, a potem wg. jego stałego „tempa” synchronizowanie aktualnej pozycji przebiegu rastra.

```
sei
lda #$35 ;wyłącz kernal i basic
sta $01

lda #$00
sta $d020;ekran na czarno
sta $d015;bez Sprites
sta $d012;pierwsze wywołanie irq

;na samej górze (linia 0)

lda $d011 ;wyłącz ekran
                (bez Badlines)
and #$6f
sta $d011

lda #$7f ;tylko przerwania
                VIC ($d012)
sta $dc0d
STA $DD0D
BIT $dc0d      ;odśwież rejestr
BIT $DD0D

bit $d011 ;czekaj na zakończenie
                ekranu
bmi *-3
bit $d011
bpl *-3

lda #$3e ;ustaw interwał
                odliczania dla zegara
sta $dc04;(63 cykle dla PAL)
lda #$00
sta $dc05

lda #$01 ;"samodocyklująca się"
                część kodu
cmp $d012
```

```
bne *-3
jsr delay_0
bne *+5
cmp $00
nop
jsr delay_1
bne *+4
cmp $00
jsr delay_1-1
beq *+2
;tu osiągneliśmy wycyklowanie...

lda #$11 ;uruchom odliczanie
                dokładnie
sta $dc0e ;w TYM momencie!

lda #<Stable_IRQ
                ;jeszcze ustawienia dla irq
sta $ffe
lda #>Stable_IRQ
sta $fff

lda #1
sta $d01a
sta $d019
cli                ;no i jazda...
```

;"neverending" pętla - symulator losowości kodu poza przerwaniem

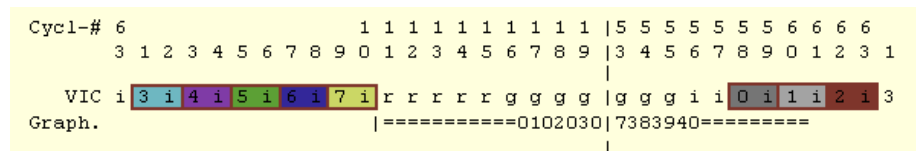
```
loop ldy $dc04
loop2 ldx $dc04
dex
bpl *-1
dey
bpl loop2
bmi loop
```

-----

```
delay_0
;procedura opóźnienia
ldx #$a2
;wykorzystana przy inicjacji
ldx #$a2
;TIMER A CIA1
ldx #$a6
delay_1 nop
ldx #$07
dex
bne *-1
lda $d012
cmp $d012
rts
```

-----

```
Stable_IRQ ;Przerwanie IRQ
pha
;wejście do przerwania
txa ;zachowaj rejestry
pha
tya
```



Pic.5

```

pha
;"wyrównaj" cykle
sec
lda #$32
sbc $dc04 ;wartość
                zmienna
sta *+4 ;dostosowanie
                do tej wartości
bpl *          ;skoku
lda #$a9
lda #$a9
lda #$a9
lda $ea
                ;tu osiągnęliśmy już
                stabilny „raster” !

;przykładowa procedura w przerwaniu
lda $d012
clc ;kolejne wywołanie
                przerwania
adc #15 ;za 15 linii
sta $d012
ldx #$0f
loop stx $d020 ;mały efekt
                kolorystyczny
dex
bpl loop
;i wyjście z przerwania
inc $d019
pla
tay
pla
tax
pla
rti

```

Jak widać w listingu, na początku wybieramy miejsce od którego ruszy nasz „zegar” – jest to tuż za ekranem na dolnej ramce – powyżej 256 linii (starszy bit pozycji rastra w *\$D011* zapalony). Nie ma tam oczywiście Badlines, a Sprite też w tym czasie powinny być wyłączone – a przynajmniej nie powinny „kręcić się” w pobliżu. Pierwsza procedura wycyklowująca, zasadą działania zbliżona jest do tej ze starych C&A – choć widać że nieco bardziej zaawansowana technicznie, ale można tu też wykorzystać metodę podwójnego przerwania. Licznik (Timer A) zostaje uruchomiony cyklicznie dokładnie w 16 cyklu linii rastra 257, i odmierza czas co 63(\$3E) cykle – jak się nietrudno domyślić, jest to czas równy wyświetlaniu jednej linii rastra w systemie PAL. Wystarczy go uruchomić raz na początku, a ponieważ każda z 312 linii rastra ma tyle samo cykli, więc nie zgubi nam już tej synchronizacji - nie mówimy tu o *Badlines* i *Sprite*, bo one mają wpływ tylko na blokowanie dostępu dla procesora, a nie wyświetlania obrazu, czy odliczania przez liczniki.

Inicjowaliśmy jeden z liczników, ale należy tu zwrócić uwagę, że nie daliśmy mu możliwości wywołania przerwania (*\$7F->\$DC0D*). Gdy odliczy do końca, to co się wtedy dzieje? Nic, licznik odlicza w kółko, ale nie miesza się do przerw, robi to niezależnie, i nie wpływa na pracę procesora. Czyli ustawiliśmy zegar – co teraz. Ustawiamy „zwykłe” przerwanie rastra, i je wywołujemy. Tu od razu rzuca się w oczy inne wejście do takiego przerwania, już nie sprawdzamy rejestru rastra (*\$D012*), a tylko młodszy bajt naszego zegara *\$DC04*. Po zachowaniu rejestrów X, Y i A, odejmujemy jego wartość od wartości *\$32(52)* – to wszystko zajmuje 19 cykli od pierwszego PHA. Czemu *\$32*, a nie *\$3E*? Jak już wspominałem wcześniej przy cyklowaniu, należy się nauczyć dobrze dodawać i odejmować. Jeśli uwzględnimy wszystkie podane wcześniej informacje, czyli: licznik odlicza zawsze od 16 cykla w linii, wejście do przerwania to 2+7, czas naszego programu w przerwaniu od instrukcji PHA do pobrania *\$DC04* to 19 cykli, czyli w sumie od „impulsu” minie 28 cykli, odejmując pozycję startową timera 16 – mamy 12 cykli – to jest właśnie ta różnica między *\$3E*, a *\$32*. Redukujemy tą różnicę zawsze tak, aby wynik zawierał się w granicach 0-6 cykli – czyli tyle ile może nastąpić rozbieżność w chwili wywołania. Potem sprytna samomodyfikacja kodu w skoku *BPL*, który w zależności od wyniku wygeneruje opóźnienie poprzez skok w odpowiednie miejsce (proponuję przeanalizować wszystkie 7 kombinacji) i po *LDA #\$EA* mamy już stabilnego rastra! Żeby nie powielać tego ciągu instrukcji (np. dla innych przerw w tej samej ramce), warto zrobić sobie zaraz potem skok do wybranego kodu przerwania, i ten skok w razie potrzeb modyfikować, a nie *\$FFFE/\$FFFF*. Takie ustanda-

ryzowanie wejścia w przerwanie zaoszczędza pamięć i konieczność przepisywania, a znacząco nie wpłynie na pozostałą część kodu wykonywanego w przerwaniu. Potem zostaje nam już tylko wyjście z przerwania, ale tego już nie trzeba opisywać.

Oczywiście, tu też musimy pamiętać o *Badlines* i *Sprites* – te pierwsze omijamy, wykonując przerwanie o jedną linię wcześniej i tworząc potem odpowiednie opóźnienie. Cykle które „kradną” Sprites musimy też uwzględnić przy odejmowaniu z *\$DC04*, odpowiednio skorygować wartość *\$32*, inaczej skok *BPL* może trafić w zupełnie niepożądane miejsce w naszym kodzie.

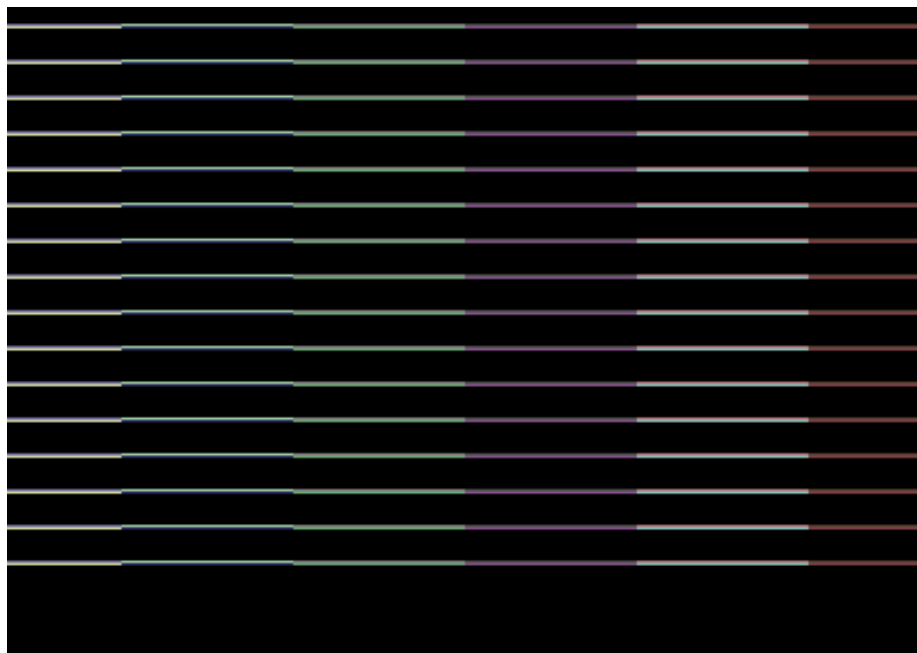
Wgryzając się w specyfikę zabierania cykli przez Sprites – okazuje się że istotne będą dla nas te z przedziału 3-7 (licząc od zera), bo tylko one zabierają cykle na początku linii rastra.

To było by na tyle z mojej strony, myślałem że opis przerwania który przedstawiłem w dość wyczerpujący sposób, oddaje istotę tego zjawiska na komodoru. I mam nadzieję, że od teraz zrozumienie pewnych rzeczy, które dzieją się w komputerze będzie jaśniejsze. No i samo programowanie nie będzie już nastroczać większych problemów lub niemiłych niespodzianek.

Temat można oczywiście jeszcze bardziej zgłębiać, ale tu odsyłam do literatury, a jeszcze bardziej do Internetu, z którego głównie czerpałem wiedzę (patrz źródła). Mile widziane spostrzeżenia i uwagi, którymi można podzielić się na forum C&A Fan lub bezpośrednio na mój e-mail.

Z komodorowskim pozdrowieniem

Skull/Samar, CA-Fan



Pic.6

# Commodore 64 GS

**Jedna z największych porażek firmy Commodore to wejście na rynek konsolowy modelem komputera Commodore 64. Jak większość pomysłów firmy Commodore ten był jednym z ich najgorszych i najgłupszych. Poprzerabiać komputer i robić z niego konsolę?**

Jednak zacznijmy od samego początku. W grudniu 1990 roku, na rynek trafia konsola *Commodore 64 Games Systems* w skrócie *Commodore 64 GS* i to akurat przed samymi świętami. Zamysł firmy Commodore był taki, że trzeba jakoś wejść na rozwijający się dynamicznie rynek konsol domowych. Pech chciał, że padło na komputer *Commodore 64*. Miał on być konkurentem pojawiających się konsol. Technicznie był to normalny C64 tyle tylko, że pozbawiony klawiatury i części interfejsów. Nawet płyta główna była taka sama jak wersji C64 G, czyli popularnej tzw. mydelniczki z białą klawiaturą, która dostała nową obudowę pozbawioną standardowych z tyłu otworów. Do kompletu dołączono firmowy joystick *Cheetah Anihilator*, który był bardzo kiepski i po dłuższym graniu nadawał się na śmietnik, zasilacz oraz kartridż z czterema grami: *Fiendish Freddy's Big Top o'Fun*, *Flimbo's Quest*, *Klax*, *International Soccer*. Jako ciekawostka gra *International Soccer* była przerobiona na użytek konsoli tak, by nie trzeba było korzystać z klawiszy funk-

cyjnych. Konsola była sprzedawana w cenie 99,99 funtów i można ją było kupić tylko w dwóch krajach w Europie. Od grudnia 1990 w Wielkiej Brytanii i od 1991 roku w Danii. Firmie Commodore udało się wyprodukować około 80 tysięcy egzemplarzy konsoli, z czego sprzedano tylko 25% z nich. Po fiasku projektu, nie sprzedane egzemplarze zdemontowano i przerobiono na komputery *Commodore 64*, które od lat dobrze się sprzedawały. Podstawowym błędem jaki zrobiła firma Commodore, to brak wsparcia peryferiów i oprogramowania. Mimo, zgodności z C64 większość oprogramowania na kartridżach działało błędnie, a to z prostej przyczyny pozbawienia konsoli klawiatury, więc 100% kompatybilności z C64 nie było mowy. Zazwyczaj ludzie, którzy decydowali się na kupno tej konsoli, korzystali z dostępnych na rynku kartridży C64, choć było ich znacznie mniej niż wydanych gier na C64. Większości kartów wydanych było z myślą o komputerze C64 i przystosowana była do nich. Posiadacze konsol po włożeniu karta do konsoli i włączeniu jej, widzieli na ekranie komunikat „naciśnij dowolny klawisz, aby uruchomić grę”. Tu zaczynał się problem, ponieważ konsola nie posiadała klawiatury, więc co mieli naciskać posiadacze tego sprzętu? Chyba najbardziej idiotyczny jest fakt, że w podobnej cenie sprzedawany był komputer *Commodore 64*, więc po co kupować taką kalekę,

słabo za te same pieniądze można mieć pełnowartościowy sprzęt. Wydanie na początku lat 90-tych konsoli z architekturą 8-bitową nie było rozsądnym posunięciem, kiedy na rynku dominowały już konsole 16-bitowe, zresztą rynek konsol był tak nasycony przez *Nintendo* i *Sega*, że wprowadzenie na niego niedopracowanej konsoli 8-bitowej z góry skazane było na porażkę. Podobno w założeniach szefostwa *Commodore* konsola ta miała konkurować nie tylko z *Segą Master System* i *NES-em*, ale z 16-bitowymi konsolami. Po drobnych przeróbkach i dodaniu klawiatury i obudowy z C64 można było z konsoli zrobić bez problemu pełnowartościowy komputer C64. Rozumiem inwestowanie pieniędzy w nowe projekty, ale nikt chyba nie zrozumie kto podjął decyzję, aby przerobić komputer na konsolę i to na dodatek gorszej niż sam oryginał. Już od samego początku tego projektu nie wrócono sukcesu handlowego, przyczynił się on tylko do osłabienia i tak już upadającej firmy *Commodore*. Obecnie wśród kolekcjonerów konsola ta jest wielkim rarytasem i bardzo trudna do zdobycia, szczególnie w pełnej, nienaruszonej wersji bez żadnych przeróbek. W tamtym okresie po kupnie konsoli wielu ludzi stwierdzało, że zrobionych w przysłowiową butelkę i starało się przerobić konsolę na komputer. W Polsce praktycznie nie do zdobycia, choć parę osób posiada ją w swojej kolekcji.

**Ramos**



# BEEPER

czyli

## mimo, że soundchipa nie mamy to gramy!

Witajcie, Nitro/Black Sun przy klawiszach. Miesiąc temu wypuściłem na Silesia Party małą produkcję o nazwie „512Beporgy” zajmującą mniej niż 1KB. Po uruchomieniu nasze oczy widzą czarny ekran, a uszy słyszą dziwną muzykę niezbyt podobną do brzmienia SID’a. Jeśli ktoś dociekliwy otworzy monitor i spojrzy na rejestry oscylatorów SID’a to zobaczy, że nie są w żaden sposób zmieniane. Więc o co chodzi? To, co słyszysz, to pierwsza na świecie odegrana na C64 muzyka beeperowa. Czym ona jest? Zapraszam do lektury!

Dawno, dawno temu, w roku 1982, w Anglii firma Sinclair Research Ltd. wypuściła swój trzeci budżetowy komputer osobisty: ZX Spectrum, jego niska cena pozwalała mu zagościć w domu nawet dość przeciętnie zarabiającej angielskiej rodziny. W stosunku do swojego starszego brata, tj. ZX81 „Gumiak” (tak pieszczotliwie nazywają Spectrumba polscy fani, ze względu na gumową klawiaturę) wzbogacił się o kolorowy chip graficzny; ZX81 nie posiadał takiego luksusu i czarno-biały obraz musiał być generowany w całości

przez procesor, co bardzo negatywnie odbijało się na wydajności. Spectrum (kolorów) w nazwie oraz tęcza na obudowie kusily potencjalnych nabywców pierwszego komputera w swoim życiu oraz użytkowników poprzednich modeli: ZX80 i ZX81.

Priorytet ceny nad możliwościami nie pozwolił, aby „Gumiak” był wyposażony w jakiegokolwiek układ dźwiękowy, konstruktorom udało się jedynie zmieścić w chipie graficznym ULA jednobitowy bzyczek (beeper) - prostsze niż on źródło dźwięku nie istnieje. Beeper potrafi przyjąć dwa stany logiczne, 0 i 1; przy stanie pierwszym oczywiście generuje ciszę, przy stanie drugim generuje maksymalną częstotliwość.



ULA – chip graficzny, który potrafi wydawać dźwięki. :)

Dźwięk z beepera może zostać skierowany równocześnie aż na trzy wyjścia, pierwszym jest wbudowany w Spectrumba głośniczek, pozostałe dwa to porty „MIC” i „EAR”, co ciekawe na porcie „EAR” dźwięk jest głośniejszy, w sam raz, aby podłączyć słuchawki – czyżby konstruktorzy przewidzieli, że niedługo po premierze komputera ktoś zacznie na beepera pisać muzykę i zechce na nich robić odsłuch?

Wystarczy na razie szczegółów technicznych, pora na kontynuację opowieści. Po wypuszczeniu „Gumiaka” na rynek naturalną rzeczą było pojawienie się gier - były one najpopularniejszym zastosowaniem wszystkich popularnych ośmiobitowców. :)

Producenci gier, aby poprawić prezentację swoich dzieł zaczęli pracować nad wydobywaniem z beepera, zamiast prostych klików, złożonej wielokanałowej muzyki oraz efektów dźwiękowych. Sztuka ta udała im się celująco - powstało wiele różnorodnych silników muzycznych oferujących (unikalne dla każdego z nich) możliwości, takie jak m.in. duża liczba kanałów (maksymalnie 8 kanałów w programie „ZX-7 Polyphonic Musical Instrument”), zaawansowana synteza (silnik Tima Follina), czy wysokiej jakości samlowana perkusja (silnik z gry „Dark Fusion” Bena Danglisha).

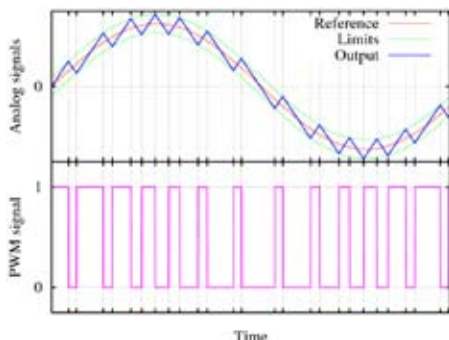
W jaki sposób odbywa się ta beeperowa magia? Kluczem do wydobywania czegoś więcej niż prostych klików z 1-no bitowego przetwornika są różne techniki, pozwalające wygenerować stany pośrednie pomiędzy zerem a jedyneką, co pozwala odegrać na beeperze np. falę trójkątną.

Omówię tutaj najprostszą syntezę, czyli PWM (Pulse With Modulation – Modulacja Szerokości Impulsu); w uproszczeniu polega ona na wykorzystaniu faktu, że beeper, generując dźwięk, nie skacze błyskawicznie z końca skali do całkowitej ciszy, tylko zajmuje to wystarczająco dużo czasu, aby procesor mógł przetworzyć nuty, wygenerować i zmiksować kilka fal dźwiękowych ze sobą oraz w odpowiednim momencie przełączyć beeper z powrotem na jedynekę/zero, co powoduje odpo-



Oto ZX Spectrum „Gumiak” – piękny, prawda? Niestety piękno często nie idzie z funkcjonalnością w parze i większość poważnych użytkowników przeklinała niewygodną klawiaturę. Wymiary: 233x144x30. Waga: 552g.

wiednią modulację generowanej fali. Brzmi skomplikowanie i jest to proces skomplikowany; wszystko musi odbywać się w idealnej synchronizacji aby dźwięk za każdym razem w miarę identycznie brzmiał; bardzo podobną sytuację mamy u nas – na C64 – kiedy musimy idealnie, co do cykla, wpisywać wartości do VIC'a, w przeciwnym razie np. rasterbary będą wesoło skakać na lewo i prawo. :) Rysunek poniżej pewnie o wiele lepiej od mojego opisu objaśni wam, jak działa synteza PWM.



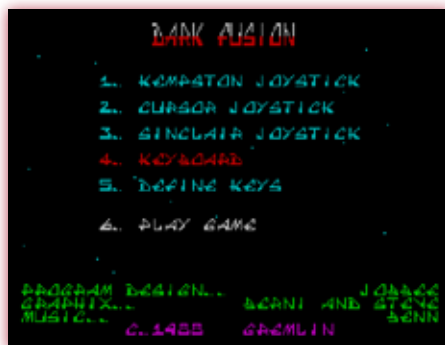
**PWM czyli coś z niczego :)**

Jak już wspomniałem, zadanie generowania dźwięku przejmuje procesor i z racji stopnia skomplikowania tego procesu nie zostaje praktycznie czasu na jakiejkolwiek czynności; możliwe jest sprawdzanie czy użytkownik wcisnął klawisz, skokowe przewijanie tekstu i kilka innych prostych efektów - to wszystko, na co można sobie pozwolić przy odgrywaniu muzyki beeperowej. Nie myślcie jednak bynajmniej, że podczas grania jesteśmy skazani na oglądanie czarnego ekranu jak to często na C64 bywa przy odgrywaniu sampli...

Chip graficzny Spectruma działa inaczej od VIC'a, tj. zatrzymuje procesor tylko i wy-

łącznie wtedy, gdy ten czyta/zapisuje obszar od 0 do 16KB, gdzie znajduje się pamięć ekranu, której nie da się relokować. Jeśli procesor wykonuje kod powyżej tego obszaru, to nie jest przez nic zatrzymywany, tak więc nie ma problemów z synchronizacją, i na ekranie za darmo możemy wyświetlać sobie obrazek. Podsumowując, muzyczki beeperowe można było usłyszeć w grach tylko i wyłącznie na statycznych ekranach tytułowych lub bardzo mało dynamicznych. Podczas rozgrywki można było usłyszeć efekty dźwiękowe, natomiast jeśli koder cechował się wysokim kunsztem programistycznym – bardzo proste muzyczki, udające najwyżej 2 kanały, a brzmiące tak (jak mówi Yerzmyey), że lepiej, aby ich w ogóle podczas gry nie było. :)

*Dark Fusion*, w tle wspaniała 3 kanałowa muzyka z świetnie brzmiącą perkusją, ale to nie wszystko! Koderowi udało się również poruszyć gwiazdki w tle i mrygać aktualnie wybraną opcją sterowania.



*Dark Fusion*, w tle wspaniała 3 kanałowa muzyka z świetnie brzmiącą perkusją, ale to nie wszystko! Koderowi udało się również poruszyć gwiazdki w tle i mrygać aktualnie wybraną opcją sterowania.

Epoka wielkiej popularności muzyki beeperowej przeminęła na wiele lat mniej więcej po roku 1988 (do roku 2006, ale to inna historia), jakieś dwa lata po wypuszczeniu na rynek komputera ZX Spectrum 128, w którym to dodano/naprawiono wiele braków „Gumiaka”, większa pamięć – 128KB, duża obudowa z solidną plastikową klawiaturą (bye bye mały zgrabny „Gumiaczek” :) ) oraz co nas interesuje najbardziej – dodano stereofoniczny trójkanałowy układ dźwiękowy AY-3-8912. Dzięki niemu proces tworzenia muzyki stał się prostszy oraz jej odtwarzanie absorbowało procesor w bardzo małym stopniu, co pozwalało wyświetlać dowolne animowane ekrany tytułowe czy też oczywiście uprzyjemniać rozgrywkę muzyką.



*Historyczny AY-Interface. Tymczasem współcześnie, znany wśród entuzjastów retro sprzętu elektronik Zaxon stworzył własną, nowoczesną wersję w 2009 roku nazwaną „AY-Magic”. Kolejną wersję interface’u, z dodanym portem joysticka stworzył Lothatek i nazwał ją „KempAY”, żartując sobie z japońskiego odpowiednika toastu – „na zdrowie!”. ;)*



*ZX Spectrum 128K, stylowy czarny design, kilkanaście nowych możliwości i wygodna klawiatura.*

Posiadacze „Gumiaków”, którzy nie mogli kupić Spectra 128, mogli zaopatrzyć się w AY-interface, tj. cart z chipem AY i cieszyć się na części grami z młodszego brata (jakkolwiek czasami okrojonymi/podzielonymi na dogrywane levele), z muzyką na AY’greku.

AY będący masowym standardem umożliwił też rozwiniecie skrzydeł demoscenie na platformę 48K: demo wykorzystujące beepera były z konieczności kolekcjami muzycznymi z obrazkami i scrollami.

Przez 10 lat scena beeperowa była praktycznie martwa... Mogę śmiało powiedzieć, że w latach 2000'ych dzięki pasji i talentcie polskiego scenowca o ksywie *MISTER BEEP*, scena beeperowa – powoli, ale stabilnie – zaczęła odradzać się z popiołów, choć był to proces trudny, bo chętni entuzjaści 1-no bitowego grania musieli zmagać się ze starymi i skrajnie nieprzyjawnymi dla kompozytorów narzędziami (jakkolwiek większość edytorów muzycznych z lat 80-tych, na wszelkie platformy, to programy „nieprzysiadalne” ; ) ).

[illegible]

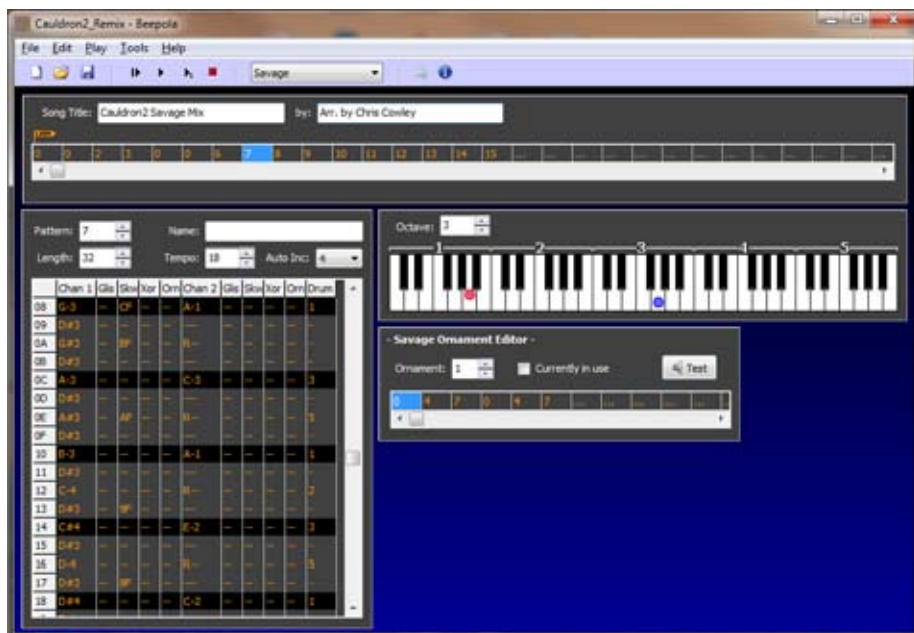
*Bądź tu mądry i pisz muzykę :)*

Prawdziwy przełom nastąpił w roku 2009'y'm kiedy to powstało kilka nowych narzędzi do komponowania muzyki na beepera o przyjaznym interfejsie trackerowym.

Trzęsienie ziemi wywołał w 2010 roku Chris Cowley, wypuszczając program o nazwie Beepola pozwalający tworzyć 1-no bitowe kompozycje pod Windows'em w bardzo przyjazny sposób. Beepola jest programem elastycznym i pozwala tworzyć muzykę pod różne silniki beeperowe, w aktualnej wersji jest ich aż siedem!

No dobrze, znacie już historię beepera na komputerze ZX Spectrum, teraz wpłynęmy na znajome wody i zobaczymy jak to 1-no bitowe granie zagrościło na komputerach z soundchipami oraz procesorami 6502 – 180 stopni różnicy w stosunku do „Gumiaka”.

Warto przedtem wspomnieć, że zjawisko muzyki beeperowej tworzonej w dawnych czasach, a nie współcześnie nie jest unikalne dla ZX Spectruma, muzyczki beeperowe



*Beepola, 'trochę' lepiej, prawda? Jeśli zechciałbyś zrobić sobie malutki urlop od komponowania na SID'zie ściągnij Beepola'ę i spróbuj swoich sił w okietznaniu jednego z siedmiu bardzo różnych od siebie silników beeperowych ;)*

powstawały też na kilka innych ośmiobitowców: radzieckie *Elektronika BK*, *Galaksija*, *kalkulatorki TI* i pewnie jeszcze kilka innych bardzo egzotycznych platform. Ale było ich bardzo mało a jakość pozostawiała wiele do życzenia.

Za ten wyczyn odpowiedzialni są dwaj scenowcy: XXL ze sceny Atari – mistrz przenoszenia softu z innych komputerów na małe Atari oraz YERZMYEY/HOOY-PROGRAM – m.in. wielki fan beepera oraz „Gumiaka”.

Okazuje się, że konstruktorzy ZX Spectrum nie są pierwszymi, którzy włożyli beeper do swoich komputerów, można w sumie powiedzieć, że zrobiła to przed nimi firma Atari! Tak, w bardzo drogim jak na czas swojego wypuszczenia komputerze z soundchipem o nazwie POKEY konstruktorzy umieścili również beepera i podobnie jak w ZX Spectrum znajduje się on w układzie graficznym, zwanym tu GTIA, który przez ponad 20 lat służył jedynie do... odgrywania kliku klawiatury! :)

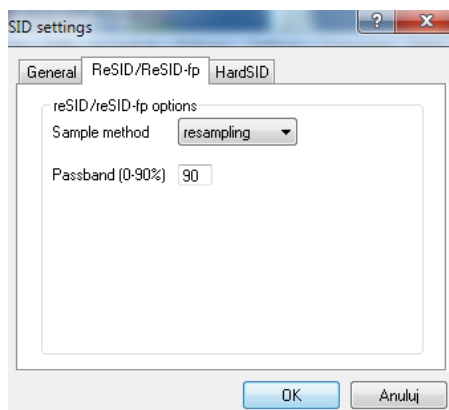
Dzięki molestowaniu...eee...motywowa-  
niu przez YERZMYEY'a, XXL zaczął przeno-  
sić procedurki odtwarzające muzyki beepero-  
we na ATARI, co na pewno było niełatwym  
zadaniem, bowiem procesor Spectruma - Z80  
- to zupełnie inna architektura oraz często-  
tliwość jego pracy jest zauważalnie wyższa,  
niż częstotliwość 6502. Aby brzmiały one  
identycznie jak na Spectrumie, XXL'a czekało  
dużo mozolnego cyklowania, z mojej strony  
wyraży szacunku! Na Atari ukazały się dwie  
kolekcje z mnóstwem muzyki beeperowej,  
przekonwertowanej z ZX Spectrum 48K: „Be-  
ep'em' All” oraz „Beep'em' All 2”.



**Kolekcja muzyczna „Beep’em All”  
na małe Atari.**

Co sprawiło, że 1-no bitowa synteza za-  
gościła również na naszym komodoru? Również przyczynił się do tego YERZMYEY: w naszych dyskusjach co pewien czas prze-  
wijał się wątek możliwości grania muzyki be-  
eperowej na C64, który przecież beepera nie  
posiada.

W pewien leniwy lipcowy poranek postanowiłem wreszcie ugryźć temat i zacząłem prace; idea była bardzo prosta: C64 może grać sample 4'o bitowe poprzez rejestr głośności, więc za jedynekę przyjmę koniec skali, czyli 15'stkę. Doksztalciłem się trochę w temacie grania sampli, umieściłem w kodzie software'owy podgłaśniacz sampli i zacząłem rozkładać na czynniki pierwsze jeden z najprostszych spectrumowych playerów beeperowych z wersji na Atari: „The Music Studio”, potrafiący grać 2 kanały i prostą perkusję. Po skopiowaniu kodu XXL'a i drobnym dostosowaniu go do realiów komodorowskich, po pierwszym odpaleniu na emulatorze usłyszałem w głośnikach mojego PC okropnie zniekształconą muzykę, która grała zdecydowanie za wolno – C64 ma niestety wolniejszy pro-



**Jeśli chcesz cieszyć się najlepszą jakością sampli i usłyszeć muzykę beeperową, koniecznie włącz „resampling” w VICE! W wersji 2.2 niestety nie działa on z „ReSIDem-fp”, wersja 2.1 jest w porządku.**

cesor niż Atari. Najgorszym problemem był wspomniany szum, obawiałem się, że inna charakterystyka SID'a może rujnować dźwięk beepera; szybkość udało mi się poprawić po prostu znajdując i zmieniając rejestr szybkości odgrywania muzyki w playerze.

Różnorakie zabiegi nie dawały niestety żadnej poprawy jakości dźwięku, zmiany silnika emulacji SID'a – również. Kiedy już miałem dać sobie spokój, otworzyłem kartę „ReSID/ReSID-fp” w ustawieniach SID'a w VICE'ie, gdzie skryła się opcja „sample method”.

W głowie zapaliło mi się światełko – u Atarowców przez dłuższy czas emulatory miały problemy z odgrywaniem muzyki beeperowej, dźwięk również charszczał: może to charszczenie jest spowodowane niedokładną emulacją sampli? Przetawiałem na metodę emulacji sampli na „resampling”, kliknąłem ok. i... zostałem uraczony komunikatem o błędzie emulacji dźwięku. :) Przetawiałem więc silnik SID'a na starszy „ReSID” i usłyszałem czysty jak łaź dźwięk beepera. Dostałem nauczkę, jeśli coś nowatorskiego na emulatorze nie działa (a powinno), to więcej niż prawdopodobne jest, że wina leży po stronie emulca.

Jeśli chcesz cieszyć się najlepszą jakością sampli i usłyszeć muzykę beeperową, koniecznie włącz „resampling” w VICE! W wersji 2.2 niestety nie działa on z „ReSIDem-fp”, wersja 2.1 jest w porządku.

Niestety z braku czasu na tym musiałem zakończyć swoje eksperymenty, a owoc mojej i przede wszystkim XXI'ej pracy, leżał sobie spokojnie na dysku, a o tym, że muzyczki beeperowe można odgrywać także na C64 wiedział prócz mnie tylko YERZMYEY.

I pewnie upłynąłby jeszcze długi okres czasu, zanim ta malutka demonstracja ujrzałaby światło dzienne, gdyby nie wspinała atmosfera na *Silesia Party 4*, która skutecznie zniechęciła mnie do kończenia na party mojego intra 4K, a pragnąc wspomóc party jakąkolwiek pracą, zamiast recyklingu moich starych efektów, wybrałem wypuszczenie tego malucha pod nazwą „512Beepergy”, 512 to rozmiar skompresowanych danych a „Beepergy” to nazwa odgrywanego utworu, którego autorem jest *Factor6/AY-RIDERS*.

I to już koniec historii, mam nadzieję, że nie zasnąłicie podczas jej czytania. :) Wystarczy teorii, czas na praktykę, zapraszam do przesłuchania choć jednego z poniższych utworów które wyselekcjonowałem dla was Yerzmyey, aby zobaczyć co potrafi beeper:

Przykładowe utwory na **Beepera** (różna ilość kanałów, od dwóch do ośmiu) -

[http://8bc.org/items/music/MISTER\\_BEEP-Greetings\\_to\\_C64.mp3](http://8bc.org/items/music/MISTER_BEEP-Greetings_to_C64.mp3)

<http://z80.i-demo.pl/mB-veeblefetzter.mp3>

[http://8bitcollective.com/items/music/MISTER\\_BEEP-Clock\\_Mechanism.mp3](http://8bitcollective.com/items/music/MISTER_BEEP-Clock_Mechanism.mp3)

[http://z80.i-demo.pl/Jason\\_C\\_Brook-Savage\\_Part\\_1.mp3](http://z80.i-demo.pl/Jason_C_Brook-Savage_Part_1.mp3)

[http://8bc.org/items/music/MISTER\\_BEEP-Nebula\\_Fight.mp3](http://8bc.org/items/music/MISTER_BEEP-Nebula_Fight.mp3)

[http://z80.i-demo.pl/Prodigy\\_Game.mp3](http://z80.i-demo.pl/Prodigy_Game.mp3)

[http://z80.i-demo.pl/Jason\\_C\\_Brook-Savage3.mp3](http://z80.i-demo.pl/Jason_C_Brook-Savage3.mp3)

[http://8bc.org/items/music/MISTER\\_BEEP-Massive\\_Implosion.mp3](http://8bc.org/items/music/MISTER_BEEP-Massive_Implosion.mp3)

<http://z80.i-demo.pl/FrankT-Fire.mp3>

[http://z80.i-demo.pl/MISTER\\_BEEP-Cosmic\\_Escape.mp3](http://z80.i-demo.pl/MISTER_BEEP-Cosmic_Escape.mp3)

[http://z80.i-demo.pl/Cobra-Magic\\_Show.mp3](http://z80.i-demo.pl/Cobra-Magic_Show.mp3)

[http://8bc.org/items/music/MISTER\\_BEEP-Binary\\_Universe.mp3](http://8bc.org/items/music/MISTER_BEEP-Binary_Universe.mp3)

[http://z80.i-demo.pl/Iredd-Phazer\\_Phunk.mp3](http://z80.i-demo.pl/Iredd-Phazer_Phunk.mp3)

[http://z80.i-demo.pl/Jason\\_C\\_Brook-Ramparts.mp3](http://z80.i-demo.pl/Jason_C_Brook-Ramparts.mp3)

[http://8bitcollective.com/items/music/Mister\\_Beep-Logos\\_Part\\_1\\_\(Tangerine\\_Dream\).mp3](http://8bitcollective.com/items/music/Mister_Beep-Logos_Part_1_(Tangerine_Dream).mp3)

[http://z80.i-demo.pl/Shadow\\_Skimmer.mp3](http://z80.i-demo.pl/Shadow_Skimmer.mp3)

[http://parishq.net/proposed/sound/zilog-jet-set\\_willy\\_cover.mp3](http://parishq.net/proposed/sound/zilog-jet-set_willy_cover.mp3)

<http://z80.i-demo.pl/FrankT-Airwolf.mp3>

[http://8bitcollective.com/items/music/MISTER\\_BEEP-A\\_crazy\\_boy\\_is\\_jumping\\_2.mp3](http://8bitcollective.com/items/music/MISTER_BEEP-A_crazy_boy_is_jumping_2.mp3)

[http://z80.i-demo.pl/1943\\_The\\_Game.mp3](http://z80.i-demo.pl/1943_The_Game.mp3)

[http://z80.i-demo.pl/Kolda\\_and\\_Mata-Miniartifici\\_\(quick\).mp3](http://z80.i-demo.pl/Kolda_and_Mata-Miniartifici_(quick).mp3)

[http://8bitcollective.com/items/music/MISTER\\_BEEP-Machines\\_Escape.mp3](http://8bitcollective.com/items/music/MISTER_BEEP-Machines_Escape.mp3)

**Chcesz więcej?** Oto kilka linków do stron o tematyce beeperowej:

**Z80 chip MUSIC SITE**

<http://z80.i-demo.pl/>

Archiwum muzyki beeperowej zgranej z realnego sprzętu do MP3.

**MISTER BEEP**

[http://mister\\_bEEP.republika.pl/](http://mister_bEEP.republika.pl/)

Współcześnie grający muzyk, używający tylko i wyłącznie beepera (MP3/dema/gry)

**BEEPER DEMOS**

[http://zxpectrum48.i-demo.pl/beeper\\_demos.html](http://zxpectrum48.i-demo.pl/beeper_demos.html)

Zestaw większości (wszystkich?) dem, używających beepera.

**1-bit music news**

<http://shiru.undergrund.net/1bit/>

Blog o 1-bitowej muzyce, często aktualizowany o informacje o nowych kompozycjach beeperowych, composach, programach itp.

**BHQ001 MISTER BEEP – “Monophonic Generator”**

[http://berlin\\_hq.micromusic.net/bhack.html](http://berlin_hq.micromusic.net/bhack.html)

Płyta Polskiego guru beepera: MISTER BEEP'a, oczywiście nagrana tylko i wyłącznie na beeperze, do ściągnięcia za darmo.

**BHQ003 MISTER BEEP – „The LOWest Fidelity”**

[http://berlin\\_hq.micromusic.net/bhack.html](http://berlin_hq.micromusic.net/bhack.html)

Druga płyta niezrównanego MISTER BEEP'a wydana stosunkowo niedawno, tym razem autor skupił się na ekstremalnie limitowanych silnikach beeperowych z których w prawdziwie demoscenowym stylu wycisnął ostatnie soki.

Zanim skończę ten artykuł, zostało mi jeszcze jedno do zrobienia... Stukrotnie podziękowania dla *Yerzmyey'a* za edukację w temacie beepera, masę informacji i sugestii, co do artykułu oraz jego korektę. Kolejne wielkie podziękowania dla *Shiru*, za wyczerpujące techniczne informacje o playerach beeperowych.

*To tyle, trzymajcie się,  
pozdrawia Nitro/Black Sun.*

# Wywiad z założycielem

## firmy **BIURO INFORMATYCZNO - WYDAWNICZE** BOGUSŁAW RADZISZEWSKI I SYNOWIE



W tym numerze magazynu główną postacią rozmowy, którą chciałbym zaprezentować większej rzeszy osób jest Bogusław Radziszewski, zanany jako założyciel firmy o nazwie Biuro Informatyczno-Wydawnicze, a także współautor *Warsaw BASIC* oraz autor kilkunastu książek o tematyce komputerowej. Oczywiście to nie wszystko, co Pan Bogusław zrobił dla *Commodore 64*, ale o tym możecie dowiedzieć się z przeprowadzonego wywiadu.

**Ramos:** W życiu prywatnych co Pan prowadzi, czy nadal związany jest Pan z komputerami?

**BR:** Pracę zawodową rozpocząłem w 1959 roku w Instytucie Podstawowych Problemów Techniki PAN (tam uzyskałem stopnie doktora i doktora habilitowanego w dyscyplinie mechanika). W 1996 roku przeszedłem do pracy w Politechnice Świętokrzyskiej w Kielcach. Tam uzyskałem tytuł profesora. Tam również na jednym z Wydziałów pełniłem funkcję prodziekana, a później prorektora ds. rozwoju kadry naukowej i współpracy z zagranicą. Pracuję aktualnie w Kielcach w Politechnice i w Warszawie w jednej ze szkół prywatnych. Poza wykładami prowadzę lub prowadziłem

m.in. zajęcia w laboratorium komputerowym (zaawansowane oprogramowanie komputerów, animacje i symulacje komputerowe procesów, projektowanie witryn Internetowych, projektowanie baz danych - SQL i PHP). Wcześniej napisałem dwa skrypty na temat budowy witryn internetowych (HTML, JavaScript, Java i PHP). Jak z tego wynika „szkoła *Commodore*” do czegoś się przydała.

**Ramos:** We wczesnych latach 80-tych posiadanie komputera graniczyło z cudem i dużym wydatkiem, a taki komputer jak *Commodore* kosztował sporo pieniędzy. Skąd u Pana w posiadaniu taki komputer?

**BR:** Kolega był na stypendium w USA i zamiast butelki whisky przywiózł mikrokomputer VIC 20. Trzeba było zdobyć jeszcze transformator 220V/110V i można było już wykonywać różne obliczenia na komputerze, który stał na moim biurku, zamiast na czterodziałaniowym kalkulatorze „Bolek i Lolek”.

**Ramos:** Jest Pan najbardziej znany ze stworzonej razem z Krzysztofem Gajewskim, najlepszej odmiany języka BASIC o nazwie *Warsaw BASIC*. Skąd pomysł na rozbudowę tego języka i jaka była historia?

**BR:** Do czasu zdobycia mikrokomputera tylko konsultowałem obliczenia wykonywane przez kolegów na większych komputerach metodą „wsadową” tzn. program zostawiało się w okienku i następnego dnia była często informacja obsługi komputera, że programu nie dało się uruchomić bo na przykład jest błąd parzystości. Mając zatem na biurku VIC 20 można byłoby wykonywać obliczenia znacznie sprawniej. Ale na przeszkodzie stał dość prosty system operacyjny tego mikrokomputera. System operacyjny był skromny, ale jak szybko okazało się był wyjątkowo podatny na rozbudowę. Ja w tym czasie nie znałem dobrze żadnego języka programowania wyższego poziomu. Zacząłem zatem od nauki języka maszynowego procesora *Motoroli 6502 (65xx)*. I w tym języku zaczęliśmy z kolegą zmuszać ten mikrokomputer do coraz bardziej zaawansowanych działań. System poznawało się przez instrukcje *PEEK*, a program wprowadzało się przez instrukcję *POKE*. Na przykład, jeśli chcieliśmy napisać w lewym górnym rogu ekranu literę „a”, to wystarczyło wprowadzić na poziomie *BASIC*a polecenie: `poke4*256,1`, lub polecenie `a9 01 8d 00 04` w języku mikroprocesora 6502. Asembler zdobyliśmy znacznie później. Powyższe polecenia w asemblerze zapisują się już bardziej przyjaźnie: `lda #$01 sta $0400`. I tak to się zaczęło. Zaczęliśmy zmuszać mikrokomputer VIC 20, a później nieco *Commodore 64* do takiej pracy, jaka była nam potrzebna. Działając na poziomie języka maszynowego czasami nawet nie wiedziałem, że w komercyjnym oprogramowaniu bardziej dorosłych komputerów nie ma takich możliwości działania jak w programikach na *Commodore 64*. Na przykład w *Warsaw BASIC*u polecenie czytania procedury zaczynało się od sprawdzenia czy nie ma tej procedury w pamięci operacyjnej i jeśli jej tam nie było, to system sięgał do pamięci zewnętrznej (magnetofon, dysk) i tam jej poszukiwał. Jeśli nie było jej na dostępnym nośniku, to system prosił o wymianę nośnika na inny. I te czynności mogły powtarzać się aż do chwili, gdy poszukiwana procedura znalazła się na udostępnionym nośniku.

Tak, tylko należy pamiętać, że pierwsze wersje były trochę bardziej ubogie. Zanim zdobyłem C 64, to rozbudowywany był VIC

20. Ten mikrokomputer miał zupełnie inne sloty niż C 64 umożliwiające jego rozbudowę. Po zdobyciu „cartridge’a” z dodatkową pamięcią VIC 20 miał chyba 24 kB pamięci operacyjnej i należało zaprojektować i wykonać „bridge” umożliwiający podłączenie kilku cartridge’ów.

Niestety VIC 20 stał się łupem złodzieja z pokoju, który znajdował się w tym samym budynku co PKO BP w Warszawie przy ul. Świętokrzyskiej. Przed kradzieżą ten mikrokomputer miał podobny system operacyjny jak WB na C 64, ale system ten był czytany z kasety.

**Ramos:** *Mimo najlepszych ocen i pochlebnych recenzji Warsaw BASIC nie był tak bardzo popularną odmianą BASIC’a jak chociażby Simons BASIC. Jakiego było zainteresowanie tym BASIC’iem?*

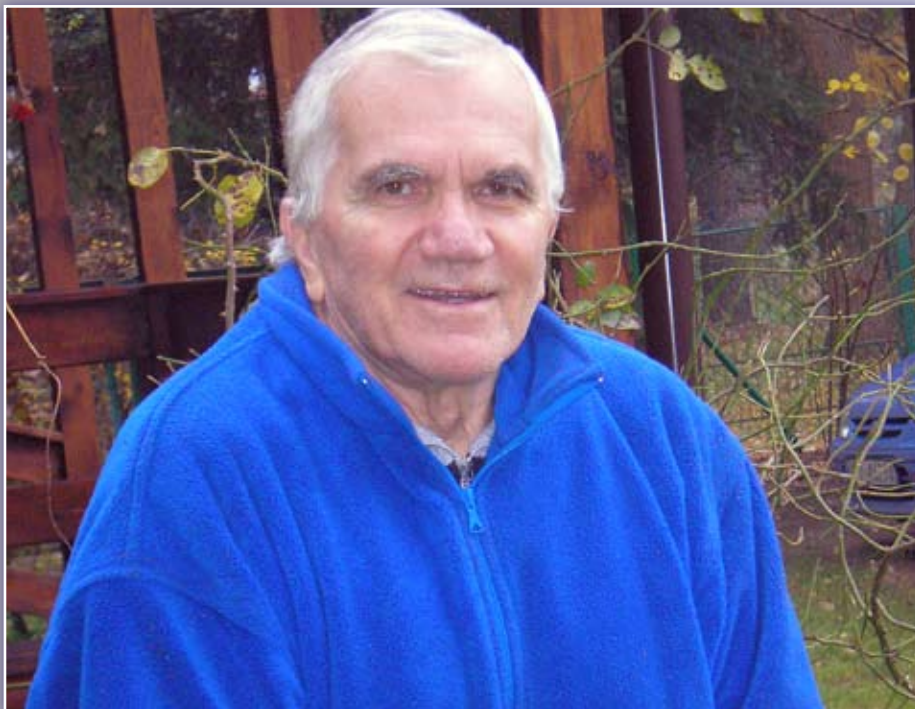
BR: Nie mam żadnej statystyki. W związku z tym mogę tylko domniemywać, że było około kilku tysięcy użytkowników różnych wersji Warsaw BASIC’a. Czasami Warsaw BASIC był niezbędnym elementem do skorzystania z innych programów takich jak cBASE (baza danych z elementami arkusza kalkulacyjnego) lub dyspozytor zadań. Miałem także kilkunastu odbiorców za granicą (Szwecja, Węgry i Czechosłowacja).

**Ramos:** *Nie wiedziałem, że tak dużo, to mnie Pan zaskoczył. Skąd w tym języku tak wiele nietypowych rozwiązań, które czasami zaskakują użytkowników Commodore’a?*

BR: Bo to był system, mam na myśli Warsaw BASIC, który był rozwijany przez kilka lat, często również pod wpływem użytkowników (oczywiście takich, którzy byli zainteresowani w tym czasie możliwościami obliczeniowymi, a nie gramami), którzy mówili „a tego to w tym systemie nie zrobić”. Stąd rozwój tego systemu nie opierał się na wiadomościach o elementach zawartych w innych językach, a na potrzebach (czasami nawet „intelektualnych”) przyjaciół i użytkowników. W większości było to pomysły autorów. Ja na przykład nie znałem dobrze w tym czasie żadnego języka programowania wyższego poziomu i kierowałem się tylko tym, by system ten był jak najbardziej przyjazny użytkownikowi (w pierwszym rzędzie autorom).

**Ramos:** *Ostatnią wersją tego języka, jak się nie mylę była wersja 3.0. Czy może miała powstać jakaś następna wersja?*

BR: O ile pamiętam ostatnią wersją był system WB 3.2. na cartridge’u. Była już wersja na tyle „doskonałą” z naszego punktu widzenia, że nie przewidywałem dalszego jej rozwoju. Tym bardziej, że system zajmował już całą pojemność EPROM’u 32 kB.



**Ramos:** *Tworząc Warsaw BASIC we wczesnych latach 80-tych takie coś jak nowy system, trzeba było znać dobrze architekturę komputera, posiadać odpowiednią dokumentację czy posiadać programy pomocne w tworzeniu czegoś takiego. Skąd w tych czasach Pan posiadał to wszystko.*

BR: Już wcześniej wspomniałem, że na początku dysponowałem mikrokomputerem VIC 20 z instrukcją użytkownika, w której był wykaz instrukcji mikroprocesora 6502. Eksploracja systemu operacyjnego polegała zatem na wykonywaniu z poziomu BASIC’a instrukcji PEEK i zapisywanie wyniku w karcie. Należy też wspomnieć, że przez kilka lat nie mieliśmy ani monitora (funkcję tę pełnił telewizor), ani drukarki, ani też stacji dysków (był tylko magnetofon). Za to było duże zauroczenie tym mikrokomputerem, gdyż praca polegająca na odkrywaniu jego tajemnic dostarczała dużo adrenaliny.

**Ramos:** *To naprawdę mrówcza praca. Obecnie nikt bez mapy pamięci C64 nie wyobraża sobie tworzenia na tym komputerze. Był Pan przez cztery lata dyrektorem Fundacji Edukacji Technologicznej (FET) czym zajmowała się ta fundacja i co miała ona wspólnego z Commodore lub Amigą?*

BR: Fundacja była założona w końcu lat 80-tych ubiegłego wieku przez grupę działaczy młodzieżowych, której działalność była wspierana przez dostawcę do PEWEX’u mikrokomputerów Commodore. Fundacja miała wspomagać sprzedaż tych mikrokomputerów w Polsce przez przygotowanie m.in. instrukcji użytkownika mikrokomputera i peryferiów w języku polskim. Sponsor przez kilka lat wspierał działalność Fundacji również finansowo. Fundacja nadwyżki

finansowo - sprzętowe przeznaczała na cele charytatywne. Organizowała również wystawy sprzętu Commodore (np. w Muzeum Techniki w Warszawie) i uczestniczyła w innych wystawach i konferencjach poświęconych technice komputerowej. Z Fundacji, po wyczerpaniu zasobów finansowych, odeszli pierwsi założyciele i propozycję kontynuowania jej działalności otrzymał Krzysztof Gajewski. I wtedy, by można było kontynuować działalność Fundacji niezbędne było zdobywanie środków finansowych na jej działalność.

W Fundacji zaczęto zatem prowadzić sprzedaż oprogramowania i literatury, początkowo oprogramowania wykonanego przez K. Gajewskiego i B. Radziszewskiego. Z biegiem czasu krąg autorów zaczął się poszerzać. Fundacja na podstawie umów zaczęła wypłacać autorom honoraria. W pewnej chwili Krzysztof Gajewski przeszedł do spółki ISR i mnie powierzono kierowanie zarządem Fundacji.

**Ramos:** *W tych czasach, kiedy w Polsce triumf święcił komputer Commodore 64, a na zachodnie rynki wchodziły komputery 16-bitowe to tworzenie na komputer 8-bitowy, z punktu widzenia zachodnich firm było bezcelowe. Jak ludzie z zachodu podchodzili na takich wystawach do prezentowanych przez Fundację zastosowań możliwości C64 do celów bardziej praktycznych niż gry?*

BR: Praktycznie dość szybko okazało się, że tworzenie czegośkolwiek na Commodore 64 czy później na Amigę było zajęciem niszowym. Sam myślałem po zdobyciu Commodore 64, że jest to w moim życiu pierwszy i ostatni komputer. Ale życie lubi płatać figle. Już pierwsze wersje WB i bazy danych zainteresowały dostawcę C64 do Polski. On

wiedział, że same komputery bez oprogramowania, instrukcji i literatury w języku polskim są produktem trudno zbywalnym. Dlatego wspierał taką działalność i dzięki temu dość szybko miałem PC'ta z twardym dyskiem około 40 MB i procesorem chyba 4.75 MHz. Inną postawę reprezentowali polscy decydenci, którzy promowali zakup jakiegoś mikrokomputera do Składnicy Harcerskiej. Oni mówili: mamy już mikrokomputer - teraz czas zająć się jego oprogramowaniem. Niestety było już za późno. Inicjatyw w tym czasie na szczeblu rządowym również nie oceniam zbyt dobrze. *Commodore 64* wraz z kilkoma innymi mikrokomputerami przedstawiałem na mini wystawie w Urzędzie Rady Ministrów jednemu z wicepremierów. Relacjonując ten fakt niezbyt dokładnie można powiedzieć, że skończyło się to podjęciem decyzji o produkcji polskiego mikrokomputera 8-bitowego we Wrocławiu w obudowie od instrumentu muzycznego, którego żywot nie był zbyt długi. Sam nieoficjalnie testowałem jeden z pierwszych polskich wrocławskich mikrokomputerów „szkolnych” i wiem ile było z nim problemów.

**Ramos:** Jak odbywało się zdobywanie oprogramowania do sprzedaży od innych twórców? Czy Fundacja gdzieś ogłaszała, że każdy kto stworzył jakieś oprogramowanie może liczyć na sprzedaż?

BR: W pewnym okresie czasu dość regularnie uczestniczyliśmy na *gieldzie przy Grzybowskiej*, gdzie były rozdawane ulotki na temat dostępnego, początkowo w *FET*, a później i w *BIW* oprogramowania, literatury i oczywiście przy okazji zapraszaliśmy potencjalnych twórców do współpracy. Jeździiliśmy samochodem „maluchem” oklejonym znakami firmowymi *Commodore* z adresem i telefonem Fundacji. Prezentowaliśmy swoje wyroby na wystawach. W Warszawie takim miejscem kilku wystaw było *Muzeum Techniki* w PKiN.

**Ramos:** Czy oprócz promocji i sprzedaży oprogramowania na *Commodore 64* odbywała się jakaś inna działalność firmy?

BR: Zaczęło się wszystko od *C64*, a później doszła *Amiga*, a już na początku lat 90-tych ubiegłego wieku również *PC*. Ale to już był schyłek działalności *BIW*. Może warto wspomnieć, że jeszcze za czasów *FET* za namową *AF (Anvar'a Fancy)* rozpoczęła się „produkcja” instrukcji i książek, początkowo na temat *Commodore 64*, a później również i *Amigi*. Były to krótkie instrukcje obsługi komputera i magnetofonu. Autorem szerszych opracowań zarówno na *C 64* jak i *Amigę* był Marek Pampuch. Były to m.in. „Pierwszy kontakt z *C 64*”, „Programowanie w *BASIC'u*”, „*Moja Amiga*” itp. Marek był autorem,

a ja redaktorem i wydawcą. Do mnie należało opracowanie tekstu pod względem „językowym”, sformatowanie tekstu na komputerze i wydruk na folii na drukarce laserowej. Niestety i tu też były problemy. By uzyskać dobrą jakość tekstu w książce należało zrobić wydruk na drukarce w postaci lustra. Aby uzyskać taki efekt musiałem zmodyfikować oprogramowanie drukarki laserowej. Do formatowania tekstu używałem wtedy program *DTP Ventura*, który wcześniej przystosowałem po pisaniu tekstów w języku polskim (lub importu z *ChiWriter'a*). Przede wszystkim należało dostosować tzw. kerning liter diakrytycznych (odstęp między literami). Jednym z atutów *FET* i *BIW* było m.in. to, że do wszystkich programów były opracowywane i drukowane (w systemie jak wyżej) instrukcje. Niektóre z nich miały postać broszury i zawierały po kilkadziesiąt stron tekstu. Mając opanowaną technikę przygotowania tekstów do druku, podjęliśmy próbę we współpracy z wydawnictwem *Ind* wprowadzenia na rynek książek dla starszych dzieci. Na pierwszy ogień poszła książka pt. *Kiki detektyw*, do której prawa autorskie miało wydawnictwo francuskie *Hachette*. Trzeba było zdobyć kolejną umiejętność, a mianowicie uzyskać z tego wydawnictwa zgodę na przetłumaczenie z języka francuskiego na język polski i druk w ściśle określonym nakładzie. Niestety w tym czasie nie było dobrych kanałów dystrybucji. Były już co prawda hurtownie książek, ale nie były one zainteresowane jedną książeczką za niezbyt wygórowaną cenę. Zwykle na wystawach w hurtowniach *Kiki detektyw* była wystawiana gdzieś w „kącie”, albo po pewnym czasie zupełnie zniknęła. Nieracjonalne próby dostarczania książki bezpośrednio do księgarni też nie przynosiło efektów, gdyż zapłatę uzyskiwało się zwykle po sprzedaży, nad którą nie mieliśmy żadnej kontroli. Książeczka ta była wydana około 1990 roku, a jeszcze dziś można ją kupić na *Allegro* w cenie od 4.00 zł do 9.50 zł (sprzedający nie ma nic wspólnego z wydawcą!). Przy okazji należałoby wspomnieć o rozpowszechnianiu książek o tematyce komputerowej. W tym czasie była w Warszawie praktycznie jedna księgarnia przy ul. Mokotowskiej (niedaleko Pięknę), która część wynagrodzenia wypłacała bezpośrednio po dostarczeniu książek. Niestety, zmniejszanie się zainteresowania mikrokomputerami *C 64* i *Amigą* doprowadziło też do znacznego ograniczenia sprzedaży literatury na ich temat.

**Ramos:** Wiadomo mi, że w 1991 w *FET* powstał kartridż o nazwie *MAX PL*. Kto był pomysłodawcą i twórcą tego cartridge?

BR: Wydaje się, że pomysłodawcą był Krzysztof Gajewski, który w tym czasie był jednym z dyrektorów *ISR*, spółką pośred-

niczącą m.in. między dostawcą z USA (ed. *ISR to International Service Resources*, a dostawca z USA to firma *Fancy Electronics Ltd.*) i *PEWEX*em w sprzedaży mikrokomputerów *Commodore* w Polsce. Projekt i wykonanie (software i hardware) spoczywało na mnie. Wykonałem kilka egzemplarzy pilotażowych, a całą produkcję miał zająć się *ISR*. Cartridge *MAX.PL* był dołączany do każdego sprzedawanego mikrokomputera. Nawiasem mówiąc nie zarobiłem na tym przedsięwzięciu tyle ile oczekiwałem, gdyż *ISR* przejął produkcję nie wywiązując się w terminie z zobowiązań finansowych. A to był przełom lat 80 tych i 90 tych ubiegłego wieku, kiedy w Polsce „szalała” inflacja.

**Ramos:** Najbardziej jest Pan znany i okarżony z firmą *BIW (Biuro Informatyczno-Wydawnicze)*. Jak wyglądała jej historia i jak to wszystko się zaczęło.

BR: Na początku lat 90-tych ubiegłego wieku, prawie cała działalność *Fundacji* przechodziła przez moje „ręce”. Przygotowanie do sprzedaży softwar'u i hardwar'u, instrukcji do nich oraz broszur i książek - kopiowanie, opracowanie edytorskie, skład komputerowy, opracowanie oBRodów drukowanych i montaż (lutowanie) na nich elementów elektronicznych o małej skali integracji, epromów (które wcześniej trzeba było zapisać odpowiednim programem), należało do moich obowiązków. Przygotowywałem również opakowania do produktów rozprowadzanych przez *Fundację*. *Fundacja* poza obowiązkiem prowadzenia działalności charytatywnej była zmuszona składać sprawozdania ze swojej działalności, zwoływać zebrania zarządu itp., nie mając żadnej pomocy ze strony sponsorów.

W tym stanie postanowiłem zamknąć działalność *Fundacji* i całą produkcję przenieść do założonych przeze mnie firm: *Biura Informatyczno Wydawnicze* - Bogusław Radziszewski i Synowie oraz do *Wydawnictwa AMBER*. Były to firmy rodzinne. Miałem dużą pomoc ze strony moich synów.

**Ramos:** Mało osób wie, że pod szyldem tej firmy oprócz wydania *Warsaw BASIC* i gier, stworzył Pan kilka programów m.in. *Edytor PL* oraz kilka rzeczy z hardware *C64*. Co w masie tych rzeczy było największym Pana osiągnięciem i z czego jest Pan najbardziej dumny?

BR: Niestety, na to pytanie trudno znaleźć odpowiedź. W większości produktów znajduje się część mego życia. Jest tutaj oczywiście *Warsaw BASIC* z całym zasobem możliwości programistycznych jak i wyprowadzaniem polskich liter (liter diakrytycznych ą, ę itp.) na ekran monitora i na papier przez drukarki dostosowane do *Commodore* (szeregowe wyprowadzanie) jak i w systemie *PC* (równoległe wyprowadzenie), *cBASE* - baza danych

z elementami arkusza kalkulacyjnego z podstawowym zasobem poleceń, typowym dla baz danych, porządkujących zasoby zgodnie z regułami języka polskiego, *Edytor PL* - ze słownikiem języka polskiego, sprawdzającym poprawność ortograficzną tekstów pisanych w języku polskim. Stąd widać m.in., że od początku staraliśmy się by z *Commodore* można „współpracować” zgodnie z regułami języka polskiego. Kiedyś uczestnicząc w wystawie i konferencji poświęconej technice komputerowej, był przedstawiany system operacyjny komputerów klasy *PDP* napisany w *PASCAL'u* i referent przeproszał, że dostępne na ten temat materiały są napisane w polskim języku komputerowym (tzn. bez liter diakrytycznych). Kolega (*K. Gajewski*) przedstawienie systemu *Warsaw BASIC* zaczął od stwierdzenia „Nasz system dotyczy komputera najmniejszego - 8-bitowego, ale materiały na jego temat są wydrukowane na tym komputerze zgodnie z regułami języka polskiego”. Zupełnie nowym przedsięwzięciem, jednym z ostatnich na ten temat, było opracowanie *Kontrolera C64* (cartridge) - programu automatycznie inicjującego system mikrokomputera *C 64* i uruchamiającego wskazane w pamięci zewnętrznej oprogramowanie po przerwie w zasilaniu. W końcu mniej „użytkowy”, ale za to sprawiający wiele satysfakcji program *Dyspozytor zadań* - system czasu rzeczywistego, który znalazł zastosowanie m.in. w laboratorium metrologicznym przy automatyzacji procesu legalizacji urządzeń elektronicznych.

**Ramos:** W tworzeniu różnych programów jak i pisaniu książek współpracował Pan z Krzysztofem Gajewskim. Jak przebiegała Pana współpraca i jak do niej doszło?

**BR:** Krzysztof Gajewski po studiach przyszedł do pracy w Instytucie Podstawowych Problemów Techniki PAN, gdzie ja pracowałem od końca 1959 roku, i po pewnym czasie przygotował pod moim kierownictwem pracę doktorską, którą obronił w 1984 roku. Nawiasem mówiąc, większość obliczeń wykonał na *Commodore VIC 20*, a wyniki w postaci zdjęć z ekranu telewizora zostały umieszczone w pracy. I to był początek prób zmuszenia mikrokomputera *Commodore* do bardziej intensywnej pracy na wyższym poziomie.

**Ramos:** Firma BIW, przez większość użytkowników *Commodore 64*, była kojarzona z tym komputerem, ale mało kto wie, że stworzył Pan programy na komputery *Amiga* i *PC*. Do czego służyły te programy i czy było na nie duże zapotrzebowanie wśród użytkowników tych komputerów?

**BR:** Autorami kilku opracowań na ten temat byli studenci, którzy realizowali swoje prace przejściowe lub dyplomowe. Progra-

my na te komputery powstały w ostatnim okresie działalności moich firm. W związku z tym krąg użytkowników nie był zbyt duży. Zresztą w tym czasie rynek komputerowy zaczął być monopolizowany przez komputery klasy *PC*. Zanim do tego doszło, to na bazie programu *Warsaw BASIC* obsługiwany był warsztat samochodowy (kalkulacje, rachunki i prowadzenie magazynu z częściami), kantor wymiany walut, pracownia geodezyjna, laboratorium metrologiczne itp. W *BIW* wydawanie rachunków i faktur, prowadzenie książki przychodów i rozchodów oraz monitorowanie stanu magazynu części, programów i książek powierzone było *Commodore 64*.

**Ramos:** Swojej ofercie *BiW* miała dużo programów, tworzonych przez miłośników komputerów *C64* i nie tylko. Czy było duże zainteresowanie produktami *BiW*? Jaka była sprzedaż i czy była opłacalna, aby utrzymać firmę?

**BR:** Może nie odpowiem wprost. Był pewien okres czasu, gdy codziennie wysyłaliśmy kilkanaście paczek (w dużej mierze była prowadzona sprzedaż wysyłkowa), czasami kilkadziesiąt. Cóż praca w tzw. budżetówce wystarczała w tym czasie na przeżycie. Dzięki moim firmom i zarobionym w nich złotych kom oglądałem transmisję z obrad Okrągłego Stołu na kolorowym telewizorze, a moja żona zaczęła pracować korzystając z pralki automatycznej firmy *POLAR*.

**Ramos:** Według użytkowników *C64* sprzedawane tam gry nie były najwyższych lotów. Jakie były kryteria przyjmowanych produkcji do *BIW*?

**BR:** Programy rozpowszechniane w Fundacji, a później w *BIW* były tworzone przez amatorów - entuzjastów mikrokomputerów. W związku z tym podstawowym kryterium było poprawne, zgodne z instrukcją ich działanie.

**Ramos:** Z tego wynika, że każdy kto stworzył ciekawy program na komputerze mógł się z Panem skontaktować i mógł liczyć na wydanie przez *BIW* swojej produkcji? Nie obawiał się Pan, że ludzie będą nie uczciwi i będą podsyłać nie swoje produkcje?

**BR:** Oczywiście, każdy mógł wydać swoje opracowanie. Wydaje się, że „podstawienie” cudzego opracowania nigdy się nie zdarzyło. Zawsze autor musiał zaprezentować swój wyrób. Ponieważ sam tworzyłem oprogramowanie, to miałem ułatwione rozpoznanie ewentualnego plagiatu. Temu celowi służyły także recenzje w *Bajtku* i *Commodore & Amiga*.

**Ramos:** Może Pan nie pamięta, ale w 1994 roku nie jaki *Zdzisław Bułka* wydał kilka chwalonych przez pismo *C&A* progra-

mów na *C64*, które okazały się plagiatami. W opisanych programach zmienił nieco szatę graficzną i wszystkie teksty angielskie na polskie. Dopiero interwencja czytelników pisma pomogła wykryć oszustwo. Dystrybutorem tych programów był właśnie *BiW*. Pamięta Pan taki incydent?

**BR:** Wydaje mi się, że takiego incydentu nie pamiętam i nie pamiętam autora o nazwisku *Bułka*. Jeśli gdzieś przewinął się taki autor, to był on o tyle nieznaczący, że nie odnotowałem tego faktu. (red. dla ciekawych odsyłam do numeru *C&A* nr 10 z roku 1994 do strony 28).

**Ramos:** Czy oprócz komputerów *Amiga*, *C64* i *PC*, myślał Pan aby zająć się dystrybucją oprogramowania na inne komputery?

**BR:** Praktycznie rzecz biorąc nigdy nie myślałem poważnie o dystrybucji oprogramowania. Mnie interesowało przede wszystkim tworzenie oprogramowania. I myślałem, że jak opracowałem coś ciekawego to powinien się znaleźć ktoś, kto zajmie się dystrybucją. I od tego zaczynałem. Ale skończyło się to niepowodzeniem. Pierwszą umowę na rozpowszechnianie wyrobów *KG&BR* (*Krzysztof Gajewski i Bogusław Radziszewski*) podpisaliśmy z firmą komputerową w Bydgoszczy (honorarium nasze miało wynosić „aż” 5% ceny sprzedaży!). Pewną liczbę naszych wyrobów dostarczyliśmy, ale pomimo interwencji ani złotówki nigdy nie otrzymaliśmy. I to była chyba główna przyczyna pomysłu o dystrybucji. Zresztą inna próba znalezienia dystrybutora też zakończyła się fiaskiem. Tym razem były to Katowice. Dystrybutor nasze wyroby brał w „komis” i zamiast je sprzedawać, wypożyczał za opłatą nasze wyroby, które użytkownik po skopiowaniu zwracał. W ten sposób pomimo obrotów „stan magazynu nie ulegał zmianie”. Zdjęcie tego dystrybutora (przy jakiejś okazji był z nim wywiad być może w *Trybunie Ludu*) przez kilka lat wisiało w naszym pokoju z podpisem „*Tęgo Pana nie obsługujemy*”. Wymienione przypadki dotyczyły programów na kasetach. W ostatnich latach działalności miałem sygnały, że zaczęto m.in. w Krakowie kopiować również cartridge. Jeszcze w czasach Fundacji w naszych ulotkach można było przeczytać „Naszą listę złodziei otwiera *OSKAR Sp. zo.o.* (patrz ogłoszenie całostronicowe m.in. a *C&A*). Firma ta kopiowała. Pierwszy kontakt z komputerem, a aktualnie sprzedaje m.in. *cBASE*” Być może, że rozpowszechnianie tego typu informacji o nieuczciwych firmach było powodem napadów na mnie kilku uzbrojonych w metalowe rurki bandytów. Dzięki sprzyjającym okolicznościom - małe pomieszczenie *BIW* i w zasięgu ręki krzesło z litego drzewa - udało się wyjść z tego zdarzenia z kilkunastoma szwami i złamanym paliczkiem. Napastnicy nic nie

mówili, także do dziś nie jestem pewien czy powodem tego była tylko moja walka ze złodziejami własności intelektualnych, których eufemistycznie nazywano w owych czasach piratami.

**Ramos: Może pamięta Pan, co było największym hitem sprzedaży z oferty produktów BIW?**

BR: W owych czasach wszystko co było opracowane w języku polskim było w jakimś sensie hitem. Ale tak bardziej poważnie mówiąc, wydaje się, że był to zestaw programów z pogranicza programów użytkowych i gier o nazwie Groch z kapustą. Autorem był entuzjasta C 64 z Białegostoku. (red. autorem tych programów jest B. Żesko)

**Ramos: W magazynie Commodore & Amiga często odbywały się recenzje produktów z oferty BIW. Jak odbywała się współpraca z tym pismem i czy miało to później wpływ na sprzedaż opisanych produktów?**

BR: Z redakcjami *Commodore & Amiga* oraz *Bajtkiem* od początku (już w czasach *FET*) miałem dobre kontakty, które pozwoliły mnie i Krzysztofowi poznać m.in. smak zawodu „dziennikarskiego”. Zobowiązaliśmy się przygotowywać do kolejnych numerów *Bajtka* serię publikacji pod głównym tytułem „Przedstawiamy Warsaw BASIC”. Inauguracja była w drugim numerze *Bajtka* w 1988 roku. Postanowiliśmy zrobić coś nowego i przystępnego dla średnio zaawansowanych użytkowników *Commodore 64*. W związku z tym, czasami oddawaliśmy materiał do redakcji w ostatniej chwili. Przedstawiliśmy w 9 kolejnych publikacjach następujące problemy: jak rozbudować interpreter?, pętla interpretera, organizacja pamięci, o przerwaniach i nie tylko, własne znaki, procedury w systemie WB, przetwarzanie wielu programów z podziałem pamięci, wywołanie procedury.

Co się tyczy recenzji wydawanych w *FET* i *BIW* programów i książek, to wydaje się, że korzyść z tego była obopólna. Informacja o naszych produktach docierała do szerokiej rzeszy użytkowników komputerów, a *Bajtek* czy *Commodore & Amiga* miały materiał, który wypełniał ich ramy. Zresztą w tych pismach ukazywały się ogłoszenia Fundacji, a później *BIW* informujące bardzo krótko o dostępnym oprogramowaniu i literaturze

**Ramos: Wszystkie produkcje zamieszczone w ofercie BIW to produkcje polskie. Zdarzyło się, aby ktoś z zagranicy chciał coś sprzedać do pańskiej firmy?**

BR: Nie pamiętam takich przypadków.

**Ramos: Wśród zamawiających produkty z oferty firmy były może osoby z zagranicy?**

BR: Kilka egzemplarzy *Warsaw BASIC*

znalazło się za granicą trochę przypadkowo, przy okazji pobytu entuzjastów C 64 w Polsce. Skrócona instrukcja do tego programu była dostępna również w języku angielskim.

Przy okazji można wspomnieć, że był również program *Edytor SU*. Był to edytor tekstu na cartridge'u całkowicie zgodny z *Edytorem PL*, ale przestrzegający reguły języka rosyjskiego - cyrylica na ekranie i na papierze ze słownikiem sprawdzającym poprawność ortograficzną tekstów pisanych w języku rosyjskim. Matryca słownika była na dyskietce i każda dyspozycja sprawdzenia poprawności tekstu była pretekstem do uzupełniania (rozbudowy) słownika o nowe poprawnie zapisane wyrazy. Jeśli pojawiał się wyraz, którego nie było w słowniku system proponował użytkownikowi dwie możliwości: włączyć taki wyraz do słownika lub pominąć. W przeciwieństwie do *Edytora PL*, rozpowszechniany z *Edytorem SU* słownik nie zawierał zbyt wielu wyrazów. Pierwsze wyrazy do słownika *Edytora PL* trafiły przy okazji przygotowania do wydania w 1989 roku „*Jak rozbudować interpreter*” przez Stołeczny Ośrodek Elektronicznej Techniki Obliczeniowej (250 stron) w postaci „ready to print”. Nie będąc zawodowymi korektorami, pomimo kilkakrotnego czytania przeze mnie i współautora Krzysztofa Gajewskiego, nasz słownik wykrył około 100 różnego rodzaju błędów (zgubione lub przedstawione litery, brak spacji, zbyt wiele spacji itp.). Sprawdzanie wszystkich tekstów doprowadziło do tego, że w ostatnim czasie słownik ten zawiera ponad 30 tysięcy wyrazów.

**Ramos: Co było przyczyną upadku firmy? Czy duża rola piractwa w tym okresie, a może mało już opłacalne tworzenie legalnego oprogramowania dla lepszych komputerów typu Amiga, PC?**

BR: W latach 1995 - 1996 stanąłem przed problemem „w co się bawić”: czy prowadzić i rozwijać swoje firmy (będąc w kręgu drapieżnego biznesu), czy też zintensyfikować pracę naukową (cisza i spokój). Wybrałem pracę naukową i zacząłem „wygaszać” działalność gospodarczą. Lutownicę zawiesiłem na kołku. Aby zbyt wcześnie nie wyprzedać moich wyrobów, ceny niektórych produktów były kilkakrotnie podwyższane. Takie postępowanie nie ustrzegło mnie jednak przed całkowitą wyprzedażą. Kilka lat temu na Allegro odkupiłem m.in. cartridge *Edytor PL*.

Piractwo znacznie wcześniej wymusiło na mnie poznanie tajemnic hardware'u. Będąc z wykształcenia matematykiem musiałem nauczyć się elementów elektroniki i automatyki. Zacząłem zapisywać programy w EPROMach i umieszczać je wraz z innymi elementami na specjalnie zaprojektowanych oBRodach drukowanych. Zamknięte w specjalnych pudełkach nazywane były cartridge'ami.

**Ramos: Ma Pan może w posiadaniu nadal komputer Commodore?**

BR: W swoim archiwum mam mikrokomputery: *Commodore PET*, *VIC 20*, *Commodore 64* (stary i nowy), *Amigę 500*, *Amigę 3000* oraz monitory, magnetofony i stacje dysków i trochę oprogramowania.

**Ramos: Dużo ludzi nadal pisze różne produkcje na komputery firmy Commodore, uważa Pan że kiedyś zainteresowanie tymi komputerami całkowicie minie?**

BR: Niestety nie znam aktualnej produkcji oprogramowania związanej z C64 i w związku z tym nie znam ich „wagi”. Ja zaczynałem w czasach, gdy nie było podstawowego oprogramowania z instrukcjami w języku polskim. Stąd zainteresowanie oprogramowaniem użytkowym, pisany w pierwszej chwili na potrzeby własne, później również na potrzeby innych.

**Ramos: Z perspektywy lat, jak Pan wspomina lata spędzone przy C64 jako programista i biznesmen?**

BR: To był piękne czasy. W młodości, w czasach ostatnich lat szkoły podstawowej i pierwszych klas szkoły średniej, budowałem sobie kryształkowe radioodbiorniki. Wszystkie elementy, poza kryształkiem, kondensatorem o zmiennej pojemności i słuchawkami były własnej roboty. Wtedy też uczyłem się lutowania elementów w oBRodach elektrycznych. Po upływie prawie 30-lat, otwierała się perspektywa podobnych działań, ale na wyższym poziomie. Wydaje się, że największą satysfakcją było to, że wielu użytkowników C 64 z własnej woli wykładało pieniądze by kupić niektóre z moich wyrobów. To była nagroda za noce spędzone przy komputerze przy poprawianiu programów pisanych w kodach maszynowych, mając ciągle nadzieję, że jeśli wprowadzę „jeszcze tę poprawkę”, to program zacznie już działać. I na takich oczekiwaniach mijał czas. Poziom adrenaliny coraz bardziej podnosił się, ale na szczęście w rozsądnym czasie udawało się wiele projektów zakończyć. Pamiętam jednak przypadek, gdy w programie *Dyspozytor* zadań udało się całkowicie osiągnąć cel po rocznych poszukiwaniach błędu. Mając to na uwadze mówiłem zamykając działalność moich firm, że nikt i nic mnie w życiu tak nie upokorzyło jak *Commodore 64*. Dlatego był to jeden z najlepszych incydentów w moim długim życiu zawodowym (pracuję już ponad 50 lat).

**Ramos: Tym ostatnim pytaniem dobiegliśmy do końca tego wywiadu. Dziękuję, że poświęcił Pan czas i opowiedział kilka ciekawych historii z dawnych lat.**

Wywiad przeprowadził Ramos

# SILESIA 4

## C64 8bit PARTY



2010  
17-19 September

CZELADŹ, SILESIA, POLAND

"4th level of Evil"

W dniach 17-19 września odbyło się w Czeladzi comodorowskie party, na którym pojawiło się kilku redaktorów naszego magazynu. Podobnie jak poprzednim razem (raport w C&A Fan Nr 5), miejscem spotkania było Centrum Rozrywki i Kultury Traffic. Hasłem przewodnim party było „4th level of Evil”. O tym, że impreza się w ogóle odbędzie było już wiadomo w trakcie (lub tuż po zakończeniu) edycji nr 3. Podobnie jak wówczas, party „doczekało się” firmowych koszulek, których autorem był – tak jak w zeszłym roku – Odyn. To wspinała pamiątka, którą kiedyś będzie można pokazać swoim wnukom. Cena wstępu na party to 50 zł, jednak jeżeli ktoś wykupił bilet wcześniej, zapłacił tylko 39,99 zł. Głównymi organizatorami imprezy byli jak zwykle Raf i Volcano.

I tym razem ludzie nie zawiedli, choć mogło przyjechać ich znacznie więcej. Z obliczeń organizatorów wynika, że było ich nieco ponad 70, co nie jest złym wynikiem jak na warunki polskich imprez komputerowych. Pierwszy dzień nie był specjalnie szczególny. Większość przybyłych rozkładała sprzęt, szła do pobliskiego baru albo do sklepów zaopatrzyć się w jedzenie i jakieś mocniejsze trunki. Tak to wyglądało aż do późnych godzin popołudniowych, kiedy na imprezę dotarła już spora liczba osób. I znowu zaczęły się powitania oraz wspomniania dawnych dobrych czasów, gdy to C64 rządziło demosceną. Dowiedzieliśmy się też, że w sobotę po południu ma odbyć się obok klubu konkurencyjna impreza, a to z okazji 150-lecia dzielnicy Czeladzi – Piaski.

W dalszej części dnia odbyła się prezentacja karty muzycznej do C64 o nazwie SwinSID, którą prowadził autor urządzenia, Swinkels. W czasie prezentacji można było poznać bu-

dowę karty, sposób jej montowania w C64 oraz sprawdzić, jak radzi sobie z odtwarzaniem muzyki z tego komputera. Słychać było opinie w rodzaju: „po co taka karta, skoro w C64 jest SID?”. Dało się usłyszeć, że emulacja to nie to samo, co oryginalny SID. Sam projekt nie powstał tylko z myślą o tym, że kiedyś zabraknie SIDów i będzie je wówczas można zastąpić SwinSIDem. Oprócz znanego już potencjału SIDA, karta ta ma również swoje własne możliwości, które można wykorzystać w urządzeniach typu MIDibox.

W piątek przybyło około 50 osób, co można uznać za duży sukces. Nie licząc Polaków przybyłych z zagranicy, tj. Bookera, Leminga, Sudiego i Splatterpunka, na party pojawił się także angielski muzyk Conrad, dla którego – jak się okazało – była to pierwsza tego typu impreza w życiu. Pod wieczór miejscowi pojechali do domów na nocleg, aby powrócić kolejnego dnia. Na dobrą sprawę impreza zaczęła się rozkręcać właśnie w sobotę, kiedy na party przybyło znacznie więcej osób.

W sobotnim dodatku do Dziennika Zachodniego – Zagłębie ukazała się wzmianka o imprezie, która wzbudziła ogólną wesołość wśród partyzantów. Przeczytajcie sami: „Wczoraj Czeladź wypełnili fani starych komputerów 8-bitowych Commodore 64. Trzecia edycja Silesia Party 4, do Centrum Rozrywki i Kultury >Traffic< przy ulicy Trznadla 1, przyciągnęła tłumy z całej Europy. Jeśli ktoś chciałby do nich dołączyć, nie ma problemu (...). W dalszej części czytamy: „Na czeladzkich ulicach pojawiła się już wczoraj zaskakująco duża liczba samochodów marki FSO, a w każdym z nich znajdował się komputer z lat 80 (...). Z tego, co mi wiadomo, miłośników FSO nie było dużo, bo raptem jedna osoba, którą był

sam organizator Raf. To pewnie jego polonez robił za tłum innych samochodów. Wynika z tego, że prasa kłamie :)



Podczas gdy jedni dopiero co przyjechali na imprezę, niektórzy – jak Komek, przedstawiciel magazynu Komoda – musieli już zmykać do domu. Jak sam mówi: „Moim zdaniem Silesia Party 4 to udana impreza, przede wszystkim fajny klimat oraz sympatyczni ludzie. Szkoda, że byłem tylko z piątku na sobotę. Mam nadzieję, że na SP 5 nadrobię zaległości”. Jako ciekawostkę podam, że Komek, chociaż był tylko jeden dzień, zapłacił za całość, jednak – jak sam podkreśla – nie ma to znaczenia.

Impreza rozkręciła się na dobre od sobotniego popołudnia, a to za sprawą czołowego zorganizowanego przez Rafa. Pierwsze tego typu compo polegało na przeciąganiu liną poloneza, w podobny sposób jak to robią na zawodach



strongmen. Chętnych do zabawy nie brakowało. Pierwszy spróbował swoich sił Data. Przetarł on szlaki innym, którzy mogli później czerpać z jego doświadczeń i podpowiedzi. Zarówno publiczność, jak i osoby biorące udział w tym konkursie dobrze się bawili. Zwycięzcą okazał się Klax, który w ciągu 11 sekund bez problemu przeciągnął poloneza. Po zakończeniu compo odbyła się projekcja produkcji scenowych z PC w trójwymiarze. Organizatorzy postarali się o okulary 3D i można było podziwiać ciekawe demka zrobione w tej technice. Gdzieś około 16.00 lub trochę później odbyło się następne crazy compo. Polegało na wypiciu w jak najkrótszym czasie 1,5 l gazowanej wody mineralnej. Wygrał Klax i było to jego drugie wygrane compo na tym party.

Jeżeli chodzi o gości na party, to dodam jeszcze, że pojawili się znajomi ze sceny amigowej: Azzorek i Kamil, organizatorzy zlotu amigowców o nazwie SAP. Zawitała nawet lokalna telewizja, TV Czeladź, która chciała nakręcić reportaż z imprezy. Kamerzysta filmujący imprezę nie chciał tak naprawdę pokazać,

w jakich warunkach bawią się ludzie, robiąc ujęcia bardzo wybiórczo. Garnki, w których gotował Warlock niestety nie pojawiają się w TV, podobnie jak butelki, które wały się po sali. Z tego wynika, że telewizja pokazuje tylko wygodne dla siebie ujęcia, nie stanowiące całej prawdy, czyli standard – telewizja kłamie :) Pani redaktor przeprowadziła krótki wywiad z Rafem, Nitro i V-12, którzy jak tylko mogli najlepiej odpowiadali na jej dziwaczne pytania. Jak się później okazało, materiał z party nie został wyemitowany, bo zabrakło dla niego miejsca i ważniejsze było pokazanie reportażu z obchodów 150-lecia dzielnicy Piaski.

Sobota to zwykle czas różnych konkursów, przyszedł więc moment rozpoczęcia kompotów. Na pierwszy ogień poszło compo muzyczne. Prac było 13, a więc o jedną więcej niż rok temu. Zaskoczeniem dla większości była wygrana Klaxa, który do tej pory na takich imprezach nie zajmował czołowych miejsc, zawsze będąc gdzieś z dala (uważni Czytelnicy z pewnością zauważyli, że to już jego trzecia wygrana na tej edycji Silesia Party). Drugie miejsce zajął czołowy muzyk polskiej sceny, Jammer, który – podobnie jak w tamtym roku – stworzył

muzykę z samplami. Pojawiła się dwójka ludzi, którzy nie byli dotąd kojarzeni z muzyką: DJ. Gruby, bardziej znany jako koder i raczej początkujący grafik oraz nowy człowiek na scenie – K7. Tym razem organizatorzy odeszli trochę od restrykcyjnego dotąd standardu kompozycji i można było na compo zaprezentować aranżacje jakiejś znanej muzyki lub po prostu cover.



Po muzyce przyszedł czas na grafikę. Na konkurs wystawiono ogółem 15 prac, tj. o 4 mniej niż rok temu. Dla wielu okazało się zaskoczeniem, że drugie miejsce na gfx compo zajął grafik z Atari, znany jako Piesiu. Jak to zawsze bywa na takich konkursach, zawsze pojawiają

się grafiki stworzone dla kawału, które mają na celu rozbawienie zgromadzonej publiki. W tym roku wygrał Carrion, który ostatnio jest jednym z najbardziej produktywnych grafików na polskiej scenie. Wysokie, bo aż 4 miejsce Sebalozu warto jest odnotowania, bo od bardzo dawna ten grafik nie był tak dobrze notowany na imprezach, w których było dużo dobrej grafiki. Nie odbyło się bez małego incydentu. Jedna praca na compo, w tym przypadku Joodasa, została dostarczona bez dołączonej wyświetlarki, ale organizatorzy poszli autorowi na rękę i dopuścili ją do konkursu, odpalając grafikę z peceta.



Po grafice przyszła pora na 4KB intra. Wystawionych zostało 5 prac, wśród których wybiły się pierwsze trzy pozycje. Wygrał bezapelacyjnie Stilgar / Albion Crew swoim produktem o nazwie I2. Drugie miejsce to 512Beeporgy autorstwa Nitro / Black Sun. W 512 bajtach Nitro pokazał, że da się na C64 odtwarzać muzykę z beepera ZX Spectrum bez użycia SIDA. Wadą tej metody jest to, że strasznie pochłania czas pracy procesora i nie ma już miejsca na nic innego. Na dodatek jest to tylko 1bitowa muzyka.

Niestety demo compo totalnie większość osób rozczarowało. Nie ma jednak co się dziwić, zważywszy że w niedługim czasie miało odbyć się party w Holandii i kilku ludzi tam się wybierało. Wiadomo, że tam bardziej warto pokazać demo swojej grupy niż u nas w kraju. Do konkursu zgłoszone zostały dwie produkcje. Pierwsza to animacja żołnierza z pece-towskiej gry Wolfenstein grupy Elyta o nazwie Wolf4k. Moim zdaniem powinna być trochę lepiej dopracowana, ale takie są już produkty grupy Elyta. Drugie demo bardziej zasługuje na uznanie. Trzeba zaznaczyć, że zostało ono ukończone dopiero na party. Grupa Tropix w swoich demach niczym nowym nas nie zaskakuje, bo nie ma w nich nowych matematycznych efektów – są to dema w starym stylu

polских produkcji z początku lat 90-tych XX wieku. Demo powstało z okazji 15-lecia grupy i to ono wygrało compo.

Trzeba wspomnieć, że było jeszcze sample compo, w którym wziął udział tylko V-12 utworem „Gdzie Jest Krzyż” i oczywiście to on wygrał. Odbyło się też organizowane tylko w Polsce 2SID, w którym wzięły udział 3 utwory. Zwycięzcą okazał się Randall.

Impreza udała się i ludzie pozytywnie się o niej wypowiadali. W sumie pojawiło się około 70 osób, które wpadły nawet na chwilę, aby móc się przywitać ze znajomymi, jak choćby MrMat. Oficjalnie zarejestrowanych było 62 gości, co jest dobrym wynikiem.

Podziękowania należą się Sebalozowi za sponsoring party oraz organizatorom, którzy stworzyli naprawdę fajną imprezę. Oczywiście podziękować trzeba także tym wszystkim, którym chciało się przyjechać, bo obecnie na tego typu imprezy przyjeżdża już mało osób.

Za rok spotykamy się znowu. Jak mówi Raf w jednym wywiadzie: „O co chodzi? Głównie o to, żeby się brać komodorowska zebrała, pokazała co potrafi i dobrze przy tym bawiła...”. Trzeba przyznać, że ten cel udało się osiągnąć w 100 procentach.

**Wasz reporter**  
**Ramos**

Więcej informacji o imprezie można znaleźć na stronach: <http://c64scene.pl/viewtopic.php?p=10745#10745>  
[http://riversedge.pl/silesia\\_party\\_4](http://riversedge.pl/silesia_party_4)

A oto wyniki z kompotów:  
- półtora litra mineral compo lub inaczej obciążanie minerała:

- 1) Klax
- 2) Data
- 3) V-12

- obciążanie poloneza:

- 1) Klax 11s
- 2) Volcano, Raf, Nightrider 13s
- 5) Kisiel 19s
- 6) Nitro 21s
- 7) Splattpunk 24s
- 8) Data 40s

#### Wyniki:

Music compo:

1. 4th Level of Evil / Klax (200)
2. Still Floateeng / Jammer (193)
3. Electro Mechanica / Jan Harries (185)
4. The Beat Bizarre / Randy (184)
5. Darling Grease / Conrad (177)
6. Frozen Nipples / Leming (156)
- Lullaby for Violaine / Randall (156)
8. Eros n Psycho / Rio (155)

9. Lampa i sofa / Surgeon (155)
10. Not the Last One / Voyager (148)
11. The Incredible Hulk Theme / DJ Gruby (128)
12. Indie / Olek (110)
13. Heniek / Vegeta (95)

2 SID compo:

1. Respublika Kazantip / Randall (196)
2. Stereo Pendejo / Booker (188)
3. Mothafucka / Surgeon (186)

Sample compo:

1. Gdzie jest krzyż / V-12 (193)

Graphic compo:

1. Waiting for Inspiration / Carrion (215)
2. Holy Pilgrims / Piesiu (207)
3. Quest / Questor (188)
4. Dithering Suux / Sebaloz (187)
5. River Night / Bimber (180)
6. Ganja Vision / Miras (174)
7. Black Tampon / Leming (159)
8. Man / Zielok (142)
9. Puszka / (141)
10. Dark Dragon / Joodas (137)
11. Children of Entropy / Uka (113)
12. Scream 2 / Kenji (105)
13. Still Alive / Raf (103)
14. Slime / Lolita (94)
15. The Unknown / Cobra (89)

4 kB compo:

1. I2 / Albion Crew (221)
2. 512Beeporgy / Black Sun (177)
3. 4kb / Miracles (172)
4. Black Hole / Data (143)
5. Chuck Norris / V-12 (106)



Demo compo:

1. No Way / Tropix (180)
2. Wolf4k / Elyta (119)





# LISTY

Dziękuję wszystkim za pochlebne opinie i wytknięcie nam kilku wpadek. Na szczęście jest ich coraz mniej niż w poprzednich numerach. Postaramy się dostosować do waszych wskazówek i z numeru na numer będziemy coraz bardziej pracować na jakością pisma. Niestety brakuje nam czasu na tyle, aby numer ukazywał się w miarę regularnie. Jeśli ktoś czuje się na chęciach, to zapraszam do pisania artykułów, bo tego nie jest dużo, a każda pomoc w postaci nowych artykułów nam się przyda.

Tym razem napłynęło kilka pytań oraz wzbudziła burzliwa dyskusja na forum magazynu, na które postaramy się odpowiedzieć.

**Czytając nowy numer PPA, znalazłem opis Państwa pisma. Zainteresował mnie jeden z artykułów i przeczytałem. Niestety w czytaniu przeszkadzał mi trochę sposób odmiany nazw własnych... Polecam lekturę: [pl.wikipedia.org/wiki/Wikipedia:Zaleceni...](http://pl.wikipedia.org/wiki/Wikipedia:Zaleceni...) z lub bez apostrofu inaczej utrudnia to czytanie, a mi jako dyslektyka dodatkowo utrudnia życie, bo później sam źle piszę. Dlatego proszę uwzględnić ten aspekt, aby nie robić krzywdy innym czytelnikom.**

Dobrze postaramy się tym zająć i przekazać reszcie redakcji wskazaną stronę.

**Proszę o artykuł dotyczący drukowania pod emulatorami C64 (Vice, CCS) z uwzględnieniem znaków specjalnych (sterujących) C64.**

Takich próśb o dany artykuł jest mnóstwo, dajcie nam trochę czasu, zanim się za to zabierzemy. Czasami trzeba poszukać informacji lub samemu rozgryźć dany użytek czy program i dokładnie go opisać, żeby później nie wyszły bzdury.

**Systematycznie wiele osób pyta, kiedy stworzymy katalog gier C64/Amiga z możliwością ich pobrania, oceny itp.**

Dużego archiwum nie stworzymy, zresztą takich archiwów o grach jest mnóstwo. Dla C64 to np. GameBase64, Amiga to np. Lemonamiga, C+4/16/116 to Plus/4 World. Te przykłady stron pisane są z pamięci, co mi się przypominało. Stworzenie takiego archiwum, to posiadanie odpowiedniej ilości miejsca na serwerze, a takiego jak na razie nie posiadamy. Inna sprawa. Kto się tym wszystkim zajmie? Stworzeniem odpowiedniej strony, segregacją czy wrzucaniem gier? Kolejna sprawa, niektóre gry mają jeszcze prawa autorskie i można mieć kłopoty za ich umieszczanie na stronie. Postaramy się zamieszczać tylko te gry i programy, które ukazały się pod szyldem grupy FanCA. Jeśli chcenie pobierać gry to zapraszamy do innych archiwów, których jest bardzo dużo w internecie.

**Po co wydawać taki magazyn, skoro w Polsce fanów komputerów Commodore czy Amiga jest bardzo mało?**

Jak twierdzi nasz redakcyjny DeTePowiec Black Light: „Jeśli nie my to kto”? Większość osób narzeka na brak czasu i przez tą wymówkę nic nie robi. Inne osoby mówią, że to nie ma sensu i po co tworzyć coś dla garstki osób. Takie rozumowanie do niczego nie prowadzi i nie warto się usprawiedliwiać, tylko trzeba coś robić i pokazać innym, że się da. Nasz magazyn, choć jest niszowym pismem, to ma ściągalność z różnych serwerów na poziomie 2 tysięcy lub trochę więcej. Jak na pismo niszowe to wcale nie jest tak źle. Okazuje się, że mało osób też wie, że jeszcze coś dzieje się w dziedzinie naszych kochanych komputerów i przez przypadek odkrywają nasze pismo, co mówi nam, że musimy postarać się o większą reklamę pisma

na sieci, aby dotarło ono do jak największej liczby osób. Założeniem magazynu jest, że mimo upływu czasu nasze kochane komputery nadal żyją i ktoś na nie coś ciekawego tworzy. Chcemy też zachować od zapomnienia różne ciekawe wspomnienia, bo ludzka pamięć czasami jest zawodna. Mamy nadzieję, że dzięki przeczytaniu naszego pisma, ktoś po latach znowu zacznie pisać programy lub napisze ciekawy artykuł. Mamy już pierwsze oznaki swojego postępowania. Magazyn „C&A” Games posłużył za inspirację do powstania nowego pisma o grach na C64, bardzo nam miło słyszeć taką informację.

**Kto tak naprawdę czyta magazyn?**

Nie prowadzimy statystyk i nie chcemy nam się tego jakoś specjalnie robić. Z docierających do nas informacji, wpisów na forum, emaili do mnie, rozpiętość wiekowa jest duża. W większości czytają nas ludzie skupieni wokół forum c64power, c64scene, ppa i kilku innych mniejszych związanych grami z Commodore / Amigi. Oprócz tych osób skupionych wokół tych forum i związanych z demo sceną, czytają nas ludzie zupełnie nie związani z żadną grupą, ani nie tworzący nic specjalnego na ten komputer, po prostu są to dawni użytkownicy tych komputerów. Chcą oni poczytać czegoś innego i ciekawego, bo nie wszyscy znają język angielski czy niemiecki. To co wychodzi w magazynie jest dla nich ciekawe i czasami dostarcza nowych wiadomości, o których nie mają pojęcia.

**Jak wyglądają prace nad nowym numerem magazynu? Jak się tworzy taki magazyn?**

Pierwsza rzecz, aby magazyn mógł działać to trzeba zawiązać zespół redakcyjny. Bez takiego zespołu, w pojedynkę szybko magazyn upadnie, poza tym wszystkie artykuły pisane

przez jedną osobę w magazynie będą jednostajne. Do tego dochodzi proszenie się o artykuły innych osób, aby łaskawie coś napisali, to na dłuższy czas kiepskie rozwiązanie. Nie ma jak stały własny zespół redakcyjny. Taki zespół powinien składać się ze: operatora DTP, korektorów tekstów, armii redaktorów i naczelnego (koordynatora całego projektu, który będzie gonił ludzi do pracy, zabiegał o artykuły itp. ) oraz Webmastera strony (człowieka odpowiedzialnego za powstanie strony internetowej pisma). Pismo bez swojej strony i forum to trochę nie na czasie, a poza tym ludzie mogą się wypowiedzieć co sądzą o magazynie. Jeśli mamy już ludzi, którzy chcą pisać, to pomysły na nowe artykuły znajdą się bez problemu, a jeśli nie to ktoś z redakcji może podpowiedzieć coś ciekawego. Naczelny w redakcji decyduje, jak ma wyglądać numer magazynu, reszta ludzi dostarcza artykuły do kogoś odpowiedzialnego za korektę. Zanim zabierzemy się za składanie pisma, trzeba ustalić przybliżony termin nadsyłania artykułów, po tym terminie następuje skład magazynu. Jednak termin terminem, a czasami zdarza się tak, że artykuły są dostarczone dopiero po nim.

***Czy oprócz wydawania magazynów, grupa ma zamiar tworzyć inne produkcje na komputery Commodore / Amiga?***

Grupa FanCA nie jest typową grupą demo i od tego są inne grupy. My jesteśmy nastawieni na podawanie informacji dotyczących komputerów spod znaku Commodore, najbardziej C64 i Amiga. Planujemy wydawać na te komputery gry, ponieważ to one przyciągają większą grupę ludzi do danego komputera. Jednak do tego celu potrzebni są pracownicy koderzy, którzy chcieliby wydawać swoje produkcje pod szyldem FanCA. Jedyny warunek wydania takiej gry dotyczy poziomu, na jakim zostaną one stworzone. Na swoim koncie mamy trzy produkcje stworzone przez Indyjr'a na C64, które mimo że powstały na programie do tworzenia gier nie są wcale takie złe. Niestety z braku etatowego koderza w grupie, nie możemy aktualnie zająć się tworzeniem gier, choć chęci i motywacja jest.

***Redakcja***



Za sprawą edytora graficznego do nowego trybu **NUFLI**, stary tryb graficzny na C64 zwany **IFLI** przechodzi już w zapomnienie. Przez wiele lat **IFLI** był traktowany za najlepszy tryb graficzny na C64. Jego największą wadą było migotanie obrazu. Obrazki nie wprawionego grafika, były nieczytelne i męczyły oczy, przez co nie można ich było oglądać. Od prawie 2 lat można już podziwiać pierwsze obrazki w **NUFLI**, ta mała galeria pokazuje jak rysują najlepsi graficy w tym trybie. Trzeba się pochwalić, że wśród nich są też i Polacy.



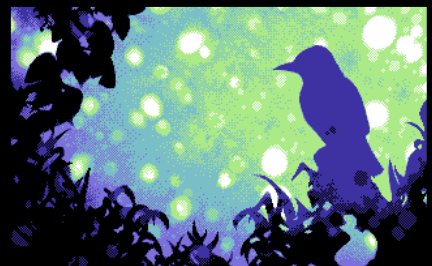
25 Years of Yie Ar Kung-Fu - by VETO



Captain' Nino's Show - by RISKEJ



D.y.m.! - by BIMBER



Dislocate - by DEEV



Don't Mess Up With Mama - by VETO



Dragon - by MIRAS



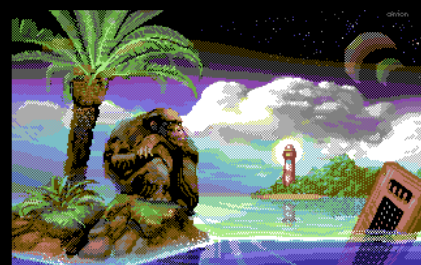
Heraklion - by CARRION



Machine B - by DEEKAY



Morswiny - by BIMBER



Waiting for Inspiration - by CARRION