

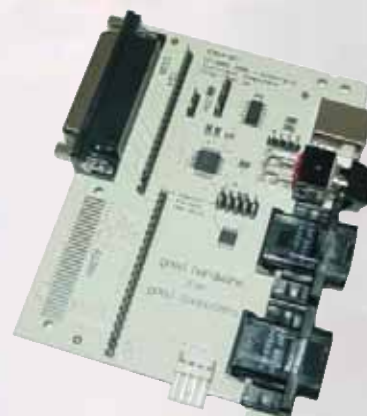
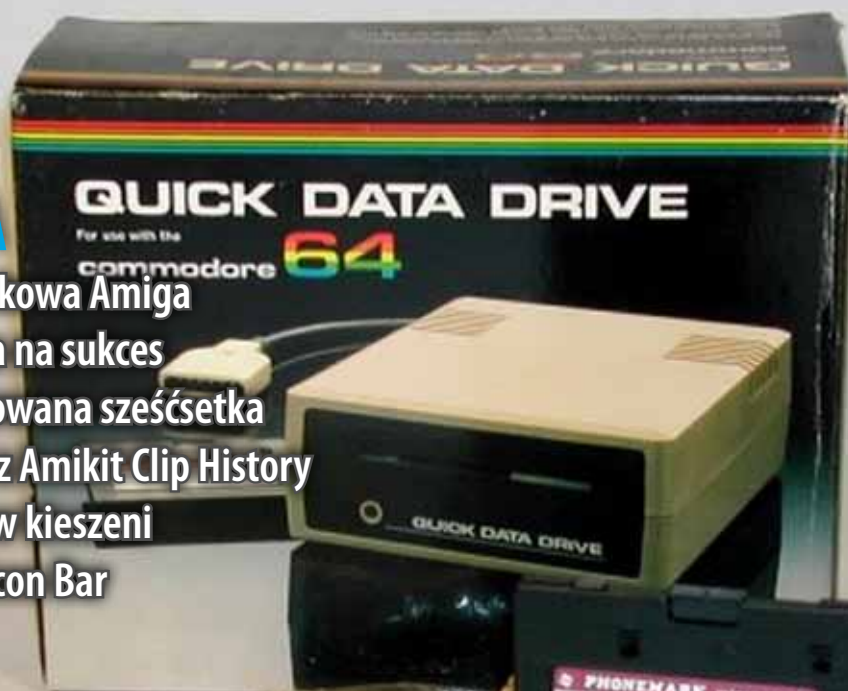
C&A fan

AMIGA

- Startrekowa Amiga
- Recepta na sukces
- Poturbowana sześćsetka
- Perełki z Amikit Clip History
- Amiga w kieszeni
- Boing Icon Bar

Commodore 64

- Quick Data Drive
- Ewolucja dem
- Moje boje z Keyrah - 8 bit
- Ratujemy Commodore 64
- Wywiad z Tomaszem Nideckim
- Adapter kasetowy
- Vintage Studio
- Commocoffe 64
- Easy Flash 3
- Loadery taśmowe cz. 1
- Róbmy Grę
- Hi-res



W NUMERZE

WSTĘP

- 2 Od redakcji
- 3 News
- 4 Rozstrzygnięcie konkursu
- 5 Jak pisać artykuły do C&A-Fan...

C64

- Quick Data Drive
- Ewolucja dem
- Moje boje z Keyrah - 8 bit
- Ratujemy Commodore 64
- Wywiad z Tomaszem Nideckim
- Adapter kasetowy
- Ventage Studio
- Commocoffe 64
- Easy Flash 3
- Loadery taśmowe cz. 1
- Róbmy Grę
- Hi-res

- 20
- 22
- 25
- 27
- 30
- 34
- 36
- 38
- 39
- 41
- 45
- 47

AMIGA

- 6 Startrekowa Amiga
- 8 Recepta na sukces
- 12 Poturbowana sześćsetka
- 14 Perełki z Amikit Clip History
- 15 Amiga w kieszeni
- 18 Boing Icon Bar

Kolejny numer CA Fan mamy już za sobą. Niestety, nie obeszło się bez problemów, które negatywnie wpłynęły przy jego realizacji. Z braku czasu i problemów w życiu osobistym chwilowo zrezygnowali ze współpracy: Indyjr, Black Light oraz arti (ten ostatni został szczęśliwym ojcem, gratulacje). Z tego powodu należało poszukać ludzi, którzy są gotowi poświęcić swój czas wolny oraz pomóc w wydaniu kolejnego numeru pisma.

Ten numer w większości poświęcony jest tematyce hardware, ale nie martwcie się, nie zabrakło także software. Miłośnicy Amigi mogą poczytać, jak uruchomić joya PC w systemie Aros, walkę Atreusa z kartą ACA 630/25, a także opis wygrzebanych perełek z pakietu AmiKit.

Miłośnicy komodorka, także nie mogą czuć się pokrzywdzeni ponieważ dowiedzą się, w jaki sposób C64 może zaparzyć kawę, jak wczytywać gry i programy na C64 bez zakupu specjalnych i drogich przystawek typu SD2IEC lub MMC Replay lub jak „ugryźć” loadery taśmowe.

Dodatkowo zostały ogłoszone wyniki konkursu z poprzedniego numeru.

Ponownie zgłaszamy się z prośbą o wsparcie redakcji w realizacji kolejnych numerów magazynu. Na przykładzie tego numeru dobitnie widać, że sami nie jesteśmy w stanie sprostać z systematycznym wydawaniem pisma.

Milego czytania oraz zadowolenia z kolejnego numeru pisma życzy grupa FanCA

Redakcja

E-Mail: Barr-find@ca-fan.pl (w sprawie strony)
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/144894602228504?v=wall>

Commodore & Amiga Fan

Magazyn użytkowników komputerów firmy Commodore

Redaktor naczelny: Ramos

Autorzy artykułów: : 11111olo, Atreus, Don Rafito, moozyk, noctropolis, Ramos, Skull, trOLLO.

Założa FanCA: arti, Atreus, Bago Zonde, Barr-find, Black Light, Don Rafito, Indyjr, MrMat, noctropolis, p.a., Ramos, Skull, TOUIdel.

Korekta: Atreus

Fotografie w wywiadzie: Tomasz Nidecki

Design, skład i łamanie: 11111olo c64.com.pl

Apel o artykuły

Szanowni czytelnicy.

Zwracamy się po raz kolejny do Was z prośbą o pomoc. Czekamy na Wasze artykuły, które zasilą nowy numer magazynu. Pozwoli to na szybsze wydawanie kolejnych numerów oraz na zróżnicowanie tematów w piśmie.

Propozycje artykułów lub Wasze pomysły proszę kierować na adres kontaktowy podany w stopce.

Newsy Newsy Newsy

Hollywood Designer 4.0

Airsoft Softwair wydał czwartą edycję programu Hollywood Designer. Program ten może być wykorzystany do tworzenia aplikacji typu: prezentacje, proste gry i programy. W tej edycji prowadzono wiele usprawnień i zmian, jednak zamiast powtarzać wszystko, lepiej zajrzeć samemu na stronę: <http://www.airsoftsoftwair.de/>



Recorded Amiga Games

Na stronie <http://www.recordedamiga-games.org/> pojawiły się nowe filmy do obejrzenia. Są to: Drip, Assimilation: Caffeine 2, Avina Blue, Gorf, Age of Rock, Das Knax Computerspiel oraz Rise of the Robots.

NetSurf v2.9

Z ainetu (<http://aminet.net/package/comm/www/NetSurf-m68k>) można pobrać nową wersję programu NetSurf w wersji 2.9. Jest to przeglądarka internetowa wspierająca HTML 4.01 i CSS 2.1.

SketchBlock v1.6



Ukazała się kolejna wersja programu graficznego pod system Amiga OS obsługującego tablety graficzne. Tym razem autor

programu Andy Broad dokonał sporo poprawek i dodał nowe funkcje. Użytkownicy do swojej dyspozycji dostali dodatkowe komendy do pracy na warstwach, narzędzie rozmazywania, rozszerzone działanie funkcji Undo/Redo i masę innych. Zapraszam do zapoznania się z programem: <http://www.broad.ology.org.uk/amiga/sketchblock/index.html>

HVSC Update #57

Miłośnicy muzyki SIDowej będą mieli nie lada gratkę, bo ukazała się kolejna aktualizacja archiwum HVSC, do którego powędrowało aż 975 mało znanych utworów. Wśród tych kompozycji będzie można posłuchać większość nie znanych kompozycji Kees Tel, brata Jeroen Tel. Oprócz tego kilka nowych kawałków stworzonych na 2SID autorstwa Conrada. Obecnie całe archiwum HVSC liczy 42212 SIDów. <http://www.hvsc.de/index.htm>

TedPlay & High Voltage TED Collection (HVTC)

Kolejne archiwum muzyczne wzorowane na HVSC. Tym razem miłośnicy rodziny komputerów Commodore 264 (C16 / C116 / +4) stworzyli podwaliny pod archiwum muzyki z ich układu muzycznego o nazwie TED. Układ ten był wyposażony w 1-kanalowy generator dźwięku oraz układ wyświetlania obrazu zintegrowany w jednej kości. W porównaniu do SIDA to żadna konkurencja.

Oprócz małego archiwum ukazał się też player wzorowany na SIDplayerze o nazwie TedPlayer.

HVTC: http://plus4world.powweb.com/features/High_Voltage_TED_Collection

TedPlay: <http://plus4world.powweb.com/tools/TedPlay>

Commodore 8bit Search

Ukazała się nowa wyszukiwarka plików do komputera C64. Dzięki temu możemy znaleźć interesujące nas pliki D64, T64, PRG oraz masę innych używanych pod emulatorami Commodore 64, a także pliki PDF. Obecnie można przeszukiwać ponad

1200 GB danych, które są umieszczone na 25 serwerach. Oprócz standardowego szukania jest opcja umożliwiająca szukanie pliku na konkretnych serwerach. <http://search.cbm8bit.com/?advanced=1>



JS VIC-20

Ukazała się kolejna wersja emulatora VIC-20 napisanego w Java przez Matta Dawsona. Emulator najlepiej współpracuje z najnowszymi przeglądarkami Chrome albo Firefox. Zapraszam do testowania jej: <http://www.mdawson.net/vic20chrome/vic20.php>

Stare Kino - Komputer Commodore

Alternatywna grupa muzyczna Stare Kino wydała piosenkę pt. „Komputer Commodore”. Lider zespołu Marcin Kasica tak tłumaczy powstanie tej piosenki: „Ta piosenka jest przewrotna. Nie chodzi w niej o tęsknotę za komuną. Wyobraź sobie taką sytuację, że twoje dzieciństwo przypomina kolorowanek dla dzieciaków. A ty masz do dyspozycji tylko czarną i szarą kredkę. Gdy w wieku dziesięciu lat uświadamiasz sobie, że paleta barw jest dużo większa zaczynasz marzyć o innych kolorach. Chłopaki z mojego trzepaka marzyli o Commodore, Atari i samochodzikach zmieniających kolory w wodzie.”

Zapraszam do przeczytania wywiadu z liderem tej grupy formacji, Marcinem Kasica: <http://www.musicarena.pl/wywiady/wywiady-pop/marcin-kasica-komputer-commodore-i-atari-xl-to-chcialem-miec/>

Teledysk: <http://www.youtube.com/watch?v=x8xFlt7xxiM>

Rozstrzygnięcie konkursu

Jak dumnie brzmią te słowa. Chciałoby się ogłosić listę osób biorących udział w konkursie, jednak jest to niemożliwe. Bezsprzecznie i definitywnie wygrał 111110lo który jako jedyny zgłosił na konkurs trzy artykuły. Otrzymuje on nagrodę w postaci książki pt. "Bajty polskie", której autorami są Bartłomiej Kluska oraz nasz red. nacz Mariusz Rozwadowski.

Ogrom prac i ilość uczestników, stawia pod znakiem zapytania celowość ogłaszania jakichkolwiek konkursów. Pierwszy który ogłosiliśmy miał być pewnego rodzaju testem, podczas którego chcieliśmy sprawdzić czy czytelnicy mają jakieś chęci do współtworzenia

gazety. Niestety wyniki mówią same za siebie. Ludzie, którzy bojkotują nas za jakieś błędy i potknięcia, a także stali czytelnicy którzy z wypiekami na policzkach oczekują kolejnego numeru, nie interesują się losami redakcji oraz napotykanymi przez nią problemami.

Były apele o pomoc, a w końcu został ogłoszony konkurs z nagrodą do wygrania. W pierwszym przypadku odzew był znikomy. W drugim prawie żaden. Taka postawa naszych czytelników zmusza do pewnych rozważań i powzięcia innych działań zapewniających jakkolwiek pomoc. Jednym z takich działań jest nowa wersja strony z wprowadzonymi na-

kazami i ograniczeniami. Jak to się sprawdzi, zobaczymy.

No, ale miało być o konkursie. Niestety, co można dalej napisać? Chyba nic i pozostawić całą kwestię milczeniem na wzór naszych czytelników. Jedyne co można dodać, to gratulacje zwycięzcy oraz jednemu uczestnikowi konkursu.

Gratulacje przesyła redakcja CA Fan.



Jak pisać artykuły do C&A Fan...

Artykuł to wypowiedź publicystyczna zazwyczaj drukowana lub elektroniczna. Do formy takiego artykułu pisanego do naszego magazynu możemy zaliczyć zwykłą notatkę o nowych wiadomościach czyli „newsy” ze świata Commodore/Amiga jak również eseje.

Ważne jest, aby każdy artykuł miał cel i prezentował coś co chcemy czytelnikowi przedstawić. Można opisać nową grę, nowy sprzęt czy przedstawić raport z imprezy komputerowej. Opisuując grę musimy nasz wybór uzasadnić, dlaczego właśnie ta gra zwróciła naszą uwagę. Każdy autor danej publikacji musi przede wszystkim wiedzieć o czym pisze i znać dany temat. Jest to wbrew pozorom bardzo ważny warunek. Kto nie zna się na tworzeniu oprogramowania nie powinien udzielać porad jak to się robi. Pamiętajmy o tym, że każdy artykuł skierowany jest do pewnej grupy czytelników. Oznacza to, że każdy tekst musi być dopasowany do poziomu ludzi którzy go czytają, więc nieraz istnieje potrzeba wyjaśnienia w tekście pewnych zagadnień lub też używanego slangu i profesjonalizmów. Jednakże nie starajmy się używać za dużo słów ze slangu czy pojęć mało znanych dla większości czytelników. Jeśli w artykule używamy określeń znanych tylko wśród wąskiego grona osób starajmy się je stosować z wyczuciem, ponieważ duża ich liczba może razić czytelnika. Trzeba też pamiętać, że dany artykuł może czytać osoba nie znająca tematu i autor powinien starać się tak napisać tekst, aby był on zrozumiały i łatwy w odbiorze. Oznacza to, że czytając nie powinniśmy co chwilę wracać do jakiegoś zdania/zdań, żeby dokładnie zrozumieć o co chodzi autorowi tekstu. Aby takich problemów uniknąć, wystarczy w naszym tekście wystrzegać się zbyt opisowych porównań i tak pisać, aby podmiot w zdaniu był łatwy do zidentyfikowania. Pamiętajmy o dobrym zastosowaniu znaków interpunkcyjnych co ułatwia czytanie. Takie znaki, szczególnie przecinek dzieli zdanie na logiczne, spójne fragmenty.

Zaczynając pisać artykuł musimy się do niego przygotować. Wybór tematu do artykułu to indywidualna sprawa każdej osoby. Trzeba jednak pamiętać, że pisząc do magazynu C&A Fan piszesz o tematyce związanej z komputerami Commodore i Amiga. Jest to duża rodzina komputerów, więc zakres tematów może obejmować pierwsze modele firmy Commodore takie jak KIM-1, PET aż po Amigi Next-genowe. To samo tyczy się opisywanych gier do magazynu o grach, przyjmujemy opisy gier stworzonych na komputery z rodziny Commodore i Amiga. Oprócz tematów związanych z tymi komputerami, możesz napisać wspomnienia jak to dawniej bywało, gdy było się użytkownikiem danego sprzętu, historie firm tworzących ten hardware, czy też historie zwykłych ludzi robiących coś dla tych komputerów lub użytkowników.

Zabierając się za pisanie artykułu należy zgromadzić odpowiednie materiały, przeczytać trochę o danej tematyce, aby później

nie podawać nie prawdziwych informacji w tekście. Pisząc artykuł pamiętajmy, że zaczyna się on od wstępu, czyli wprowadzenia w temat za pomocą podania krótkiej informacji o tym, czego dotyczy nasz artykuł. Upewnij się też, szczególnie podczas pisania recenzji gry, czy już nie została ona opisana lub ktoś inny o niej nie pisze. To samo tyczy się raportów z imprez komputerowych i innych pokrewnych tematów. W tym celu najlepiej jest skontaktować się wcześniej z Redaktorem Naczelnym Ramosem. W ten sposób zarezerwujesz niejako temat dla siebie i oszczędzisz sobie i innym czas i nerwy, gdy okaże się, że artykuł taki już się ukazał lub że są dwa na ten sam temat do jednego numeru. Następnym etapem pisania to rozwinięcie tematu artykułu.

Jest to główna część artykułu i zarazem najważniejsza. Po wyczerpaniu wątku głównego musimy pamiętać o zakończeniu, czyli podsumowaniu tego co napisaliśmy. Niech to będzie krótka puenta. Nie należy przesadzać z rozmiarami i jeszcze raz powtarzać zawartości głównego wątku (nie streszczać go). Kiedy skończymy pisać nasz tekst, nie należy od razu go do nas wysyłać. Powróćmy do niego po jakimś czasie np. na następny dzień i dokonajmy korekty. Po tym czasie wystarczy przeczytać go raz lub dwa, a niekiedy trzy, aby go poprawić bądź uzupełnić. Czasami zdarza się tak, że dopisujemy różne pominięte informacje, gdyż wydają nam się dopiero teraz istotne dla całości artykułu. Proszę korzystać także z korekty automatycznej, dostępnej w dowolnym edytorze tekstu. Jeśli macie kogoś kto może wam pomóc w merytorycznych aspektach tekstu, poproście go, aby z Wami przejrzał tekst.

Pamiętaj aby w tekście nie było zbyt rozbudowanych i zawiłych zdań. Staraj się nie używać niezrozumiałych lub fachowych terminów, bez wyjaśnienia ich znaczenia. Czasami pomagają głosem przeczytanie tekstu, aby przekonać się gdzie są błędy lub niedociągnięcia.

Krótkie zasady pisania artykułu:

1. Nie pisz tekstu sugerując się innym. Nie pisz schematycznie, bo to nie matematyka. Twój artykuł to ty sam i ty tworzysz jego niepowtarzalny styl.
2. Nie pisz pod przymusem, nie musisz koniecznie czegoś napisać. Staraj się pisać kiedy masz wenę twórczą, czas i odpowiednie warunki do pisania. Wykorzystaj tę chwilę zapału i chęci do stworzenia czegoś ciekawego.
3. Staraj się swoim tekstem zainteresować odbiorcę. Artykuł jest wtedy uważany za dobry, gdy sporo osób go przeczyta.
4. Język i treść dostosuj do odbiorcy.
5. W tekstach nie wymyślaj się i nie używaj trudnych słów, bo nikt tego nie przeczyta gdy nie będzie rozumiał tematu.
6. Nie udowadniaj na siłę swoich racji, staraj się podawać przykłady na poparcie swoich tez.
7. Kiedy opisujesz jakieś wydarzenie, to opisuj je z boku i nie staraj się go oceniać. Ocenianie

zazwyczaj należy do czytelnika.

8. Ilość tekstu nie przekłada się na jakość.
9. Pisz jasno, aby czytelnik nie musiał się domyślać o co lub o kogo chodzi w artykule.
10. Pamiętaj, że jak coś opisujesz i przytaczasz źródła to podaj je na końcu artykułu.
11. Nie zapomnij o tytule artykułu i pamiętaj aby na koniec tekstu złożyć podpis – może być ksywa.
12. Po napisaniu tekstu przeczytaj go i zrób jego korektę.

Artykuły w magazynie.

Jeśli chcesz pisać, a czujesz że dobrze Ci to nie wychodzi nie martw się, postaramy się Ci pomóc. Każdy artykuł nadesłany do nas i tak idzie do korekty, w celu wyeliminowania błędów, więc nie martw się tym zanedo. My również popełniamy trochę błędów, które można zauważyć w magazynie, pamiętaj nikt nie jest doskonały i czasem coś przeoczy. Może to być podczas korekty lub składania samego pisma. Kiedy się już zdecydujesz na napisanie artykułu i stwierdzisz, że masz ciekawy temat do przedstawienia, ale masz problem z jego napisaniem również pomożemy - od tego jesteśmy. Każda osoba ma inny styl pisanie i inaczej pisze o pewnych rzeczach i to urozmaica magazyn. Czym więcej osób będzie pisać artykuły tym magazyn będzie się częściej ukazywał i będzie w nim poruszane bardziej różnorodne tematy.

Wysyłając do nas, czyli do redakcji, artykuł należy go spakować najlepiej w formacie .zip albo .rar. Pisać można na prawie dowolnym edytorze, który jest dostępny na necie. Może być w formacie txt lub spod dowolnej wersji Worda. Jeśli do artykułu dołączamy jakieś fotografie, obrazki z gier itp. to należy dołączyć je osobno do archiwum. Nie dziel tekstu na łamy oraz nie umieszczaj grafiki, gdyż w finalnej wersji może to być inaczej niż założyłeś. Wystarczy informacja w tekście aby w tym miejscu umieścić daną grafikę (zdjęcie). Często do zdjęć są podpisy. Jeśli tego wymagasz napisz ich treść najlepiej w osobnym pliku txt lub od razu odpowiednio nazwij plik graficzny. Staraj się wysyłać zdjęcia w maksymalnej rozdzielczości – nie ma sensu ich zmniejszać gdyż negatywnie odbije się to na ich jakości. Rzuty ekranowe najlepiej zapisać w bezstratnym formacie png, tif lub ostatecznie bmp. Jpeg proszę używać tylko do zdjęć.

Na koniec pozostała jeszcze jedna sprawa. Prosimy o nie przysyłanie do Nas artykułów, które były już wcześniej publikowane, czy to w sieci lub też w innych periodykach elektronicznych bądź papierowych. Jeżeli zdecydujesz się wysłać do nas Swoje artykuły wymagamy abyś powstrzymał się od jego publikacji (lub jego fragmentów) do momentu ukazania się kolejnego numeru C&A Fan z Twoim artykułem. Po tym zdarzeniu możesz swój artykuł publikować gdzie tylko chcesz.

Redakcja

Startrekowa Amiga

Każdy ma czasami ochotę coś sobie pomajstrować, pozmienić albo chociaż zrobić coś innego. Ja mam tak samo jeżeli chodzi o komputery, dlatego dalej uwielbiam swoją Amigę. Lubię w niej pomajstrować, coś pozmienić, a przede wszystkim podoba mi się Amiga ponieważ system pozwala na dużo. Dlaczego więc nie zmienić systemu? Standardowy wygląd może nie jest zły, ale ja nie lubię standardów - rys. 1.

Jestem maniakiem Star Treka, więc stwierdziłem że chcę mieć taki sam system jak w serialu, a przynajmniej najbardziej zbliżony do serialowego.

Okazało się, że na Amidze bez większego problemu można tak przerobić



Rys. 2



Rys. 3

AmigaOS 3.9 aby przypominał system Star Treka - rys. 2 i 3.

Co będzie potrzebne ?

- Amiga – wiem że to oczywiste, ale nie było jaka, min. CPU 68040-40 Mhz, 32MB Ram, jako niezbędne minimum, kości AGA wystarczą, ale najlepsza będzie karta graficzna, 800x600 w 256 kolorach to minimum jeżeli chodzi o dobrze wyglądający system. U mnie po wielkich trudach i bojach z ustawieniami, system działa w 1280x720 przy wykorzystaniu 64000 kolorów (paleta 16bit) i przy takich ustawieniach wszystko wygląda idealnie .

Programy – do zabawy z systemem wykorzystałem:



Rys. 1

- Personal Paint 7.1, mam go z płyty CD nieistniejącego Magazynu Amiga – programem tym robię sobie tła, rysuję ikony itd.

- IconEdit – dostępny w systemie AmigaOS3.9 – ten program pozwala na stworzenie ikon dla systemu, które wcześniej rysowałem w PPaint.

- AnimatedIcon – program z systemu AmigaOS 3.9, wyświetla animowane gify na blacie systemu.

- Noisome – odgrywa różne dźwięki np. przy wkładaniu dyskietki, płytki CD itd.

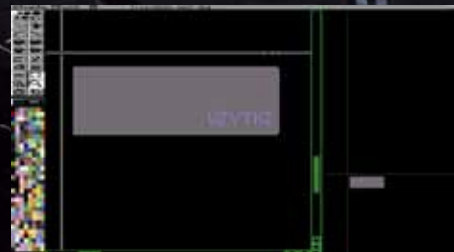
- TinyMeter – pokazuje różne informacje systemowe typu ilość wolnej pamięci RAM, obciążenie procesora itd.

Wiemy już co będzie potrzebne, więc zabawę czas zacząć.

Jak się rysuje obrazki w PPaint tłumaczyć chyba nie muszę? Wy tłumaczę za to jak robię ikonki, bo z tego co wiem, to nie każdy wie „jak to ugryźć” . Będziemy potrzebowali



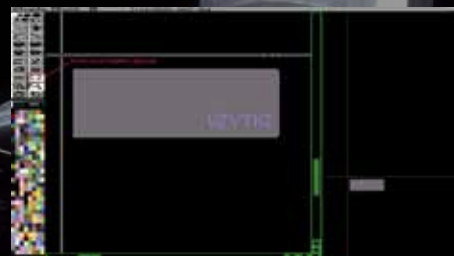
Rys. 4



Rys. 4a

PPaint oraz IconEdit.

W PPaint rysujemy swoją ikonę (rys 4 i rys 4a), zaznaczamy ją (rys 5), naciskamy F6 (F6 – zapisz wycinek – zapisze nam do pliku to co właśnie zazaczyliśmy), zazwyczaj zapisuje sobie do RAM DISK, ale każdy może zapisać gdziekolwiek indziej. Następnie, otwieramy program IconEdit, w menu jest opcja TYP,



Rys. 5

wyberamy tam, czym ma być nasza ikona – DYSKIEM, KATALOGIEM itp. Zrobimy ikonę dysku, więc wybrałem opcję DYSK (rys 6), następnie wybieramy MENU -> Obrazek ->

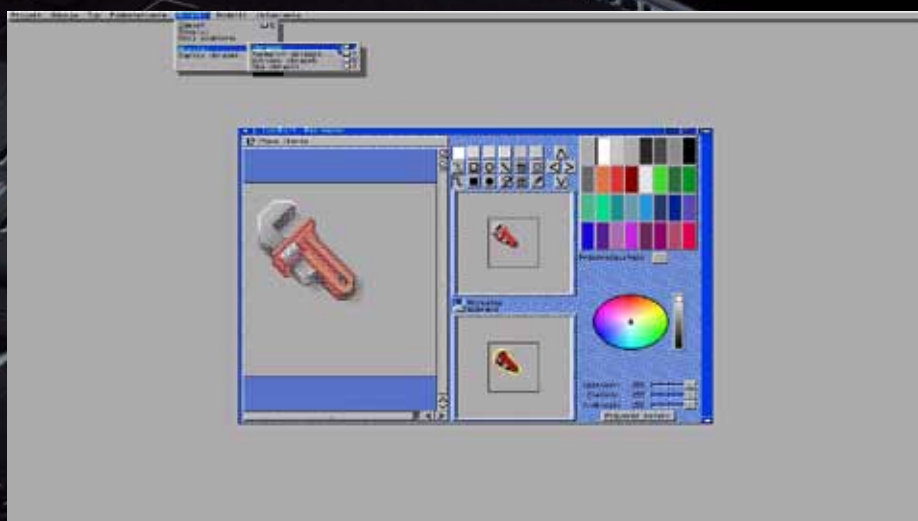


Rys. 6

Wczytaj -> Obrazek (rys 7), wczytujemy nasze obrazki dla ikony normalnej i zaznaczonej, potem dajemy zapisz i mamy gotową ikonkę.

Po dłuższej chwili system będzie wyglądał tak jak tego chcemy. W moim przypadku jak widać na obrazku (Rys 8) system wygląda jak trekowy LCARS, są też zaznaczone programy które wyświetlają moje animacje itd.

Na początku pisałem że „mój system” wymaga min. 68040/40 MHz. Dużo, ale te wszystkie animacje, muzyczki, tła itd. troszkę wymagają i coś słabszego po prostu może nie podolać. Oczywiście nie jest tak, że system „zabiera” całą moc procesora dla siebie. Na mojej obecnej konfiguracji, bez najmniejszego problemu gram w Quake I, Napalm, Exodus, DoomII i tą listę mogę jeszcze trochę wymienić.



Rys. 7



Rys. 8

Cały system który u mnie widać został zrobiony w niespełna dwie godziny. Oczywiście nie rysowałem wszystkiego sam, tła znalazłem w internecie, musiałem je jedynie dopasować do moich potrzeb. Czy warto? Jak ktoś nie lubi standardów jak ja, lubi ludzi zaskoczyć, to jak najbardziej warto. Widok prawie 20 letniej Amigi z takim systemem powoduje opad szczęki u każdego kto przychodzi. Prawie zawsze muszę otwierać obudowę i tłumaczyć, że procesor ma tylko 40 MHz, że to Amiga nie PC. Kolejnym szokiem jest karta grafiki na PCI... ale to temat na osobny artykuł.

Aleksander „trOLLO” Giedyk

Recepta na sukces, czyli moje boje z joyem w Arosie

Rozwój Arosa ciągle posuwa się na przód. Nowe wersje dystrybucji wyrastają jak grzyby po deszczu. Do tego jeszcze trzeba doliczyć uaktualnienia, lub zupełnie nowe zastępy programów i gier.

A no właśnie gier.

Ale samo granie to nie wszystko. Liczy się też sposób, korzystania z uciech teje intelektualnej rozrywki.

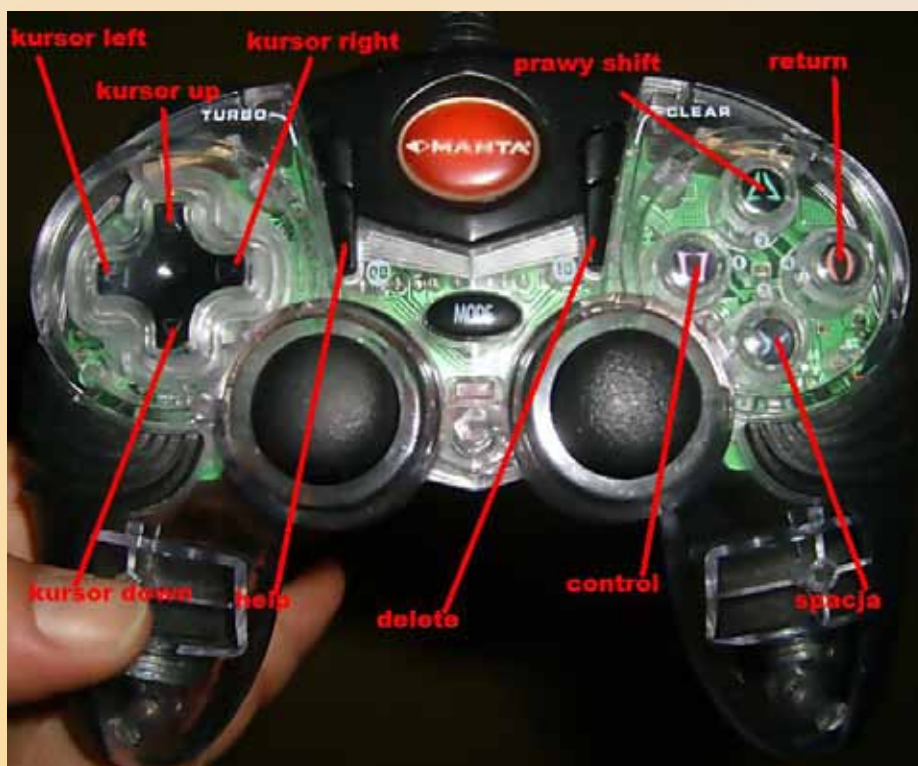
Wiadomo, podstawowym narzędziem, nie tylko przeciętnego gracza jest mysz i klawiatura. Ale historia pokazała, że joystick (czy też joypad, jak kto woli) również ma tu wiele do powiedzenia. I nie mam na myśli konsol, bo to nie ta bajka.

W Amidze klasycznej, joy jest stawiany na równi z myszką czy klawiszami. A co z resztą amigowych systemów, m. in. Arosem? Tutaj wiadomo, klasycznego kontrolera raczej nie użyjemy, za to jesteśmy w stanie skorzystać z mnogiej ilości urządzeń wpinanych w port usb, a to dzięki *Posejdonowi* i *lowlevel.library*.

Biblioteka ta właśnie odpowiedzialna jest za obsługę wszelkiej maści padów. Jeśli programista pisząc grę zaimplementuje wykorzystanie jej w swoim kodzie, sprawa jest rozwiązana. Komplikacje zaczynają się, gdy autor gry pójdzie na łatwiznę i pozostanie tylko przy podstawowym narzędziu, czyli klawiaturze. Ale Twórcy stosu usb, pomyśleli też o tym, dając możliwość użytkownikowi przypisania konkretnych klawiszy klawiatury do konkretnych przycisków joysticka.

Ale zanim zaczniemy ową konfigurację, proponuję przelecieć się po kilku grach, niekorzystających z *lowlevel.library* i zobaczyć jakie klawisze są najczęściej wykorzystywane, co też uczyniłem. Stwierdzenia są dwa. Najczęściej, dla gracza pierwszego domyślnymi klawiszami były: **kursory**, **RAlt**, **RShift**, **Ctrl**, **spacja** oraz **enter**. Drugim stwierdzeniem jest fakt, że większość gier oferowało zmianę ustawień klawiatury. To natomiast przyda się, jeśli zechcemy skonfigurować joya dla drugiego gracza.

Kolejnym etapem, jest wzięcie do ręki swojego pada i zrobienie listy, jakie klawisze zechcemy przypisać klawiaturze. Na szczęście dzisiejsze pady raczej dają możliwość wykorzystania więcej niż jednego przycisku fire, więc tutaj mamy już na samym starcie sprawę nieco ułatwioną.



Nie owijając dalej w bawełnę, przejdę do konkretów, prezentując sposób skonfigurowania joya na autopsyjnym przykładzie.

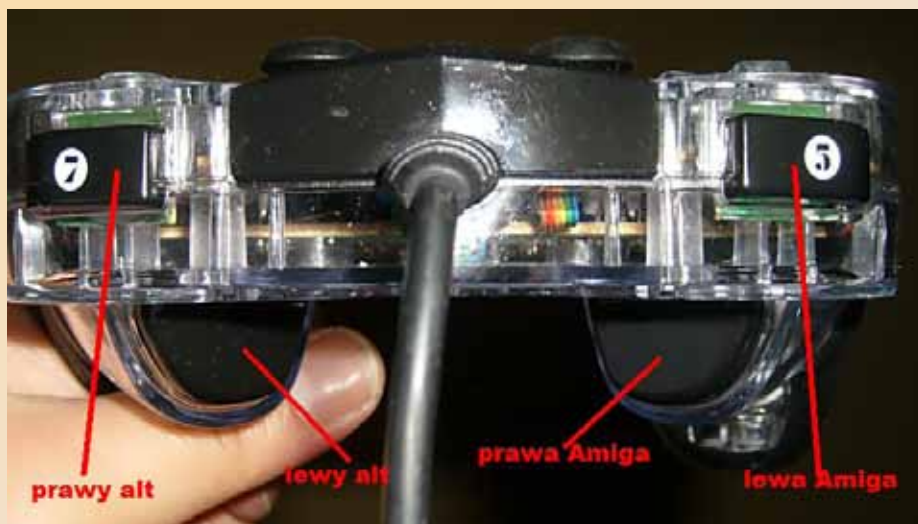
Po krótkich oględzinach, zrobiłem listę którą poprzedzona jest poniższymi zdjęciami:

Czyli:

Joy góra – strzałka kursora w górę;
Joy w dół – strzałka w dół;
Joy lewo – to wiadomo kursor w lewo;
Joy w prawo – nie inaczej jak strzałka w prawo;
Joy 1 – Control;

Joy2 – prawy Shift;
Joy3 – spacja;
Joy4 – return (enter na głównej, nienumerycznej);
Joy5 – lewa Amiga;
Joy6 – Prawa Amiga;
Joy7 – prawy Alt;
Joy8 – lewy Alt;
Joy9 – Help;
Joy10 – Delete.

Gdy mamy już przygotowane menu i składniki, przystępujemy do gotowania...





Sorry, nie ten program. Uruchamiamy komputer, po załadowaniu Arosa wpinamy pada w usb. Na blacie pojawi się komunikat *Posejdona* o wykrytym urządzeniu.

Uruchamiamy *Tridenta* (program preferencyjny *Posejdona*).

Na liście w lewej górnej części okna, klikamy na **Devices**. Po prawej stronie pojawi się nam spis, wszystkich urządzeń wpiętych w usb. Z listy wybieramy nasz joystick w dolnej prawej części listy klikamy na **Settings**.

Otworzy się nam okno preferencji *Hid klasy*, z której wybieramy trzecią zakładkę, czyli **Action**. Naszym oczom ukaże się okno z trzema listami. Górna lewa, przedstawia funkcje joja (pada). Nas interesuje tylko pierwsza pozycja z góry. Po jej wybraniu, na drugiej liście po prawej wyświetlą się nam wszystkie dostępne klawisze, począwszy od osi

Klawisz joja	Opcja klawisza	Performed actions	Trigger	Opts:	Pre-condition code	Wartości left i right constant	Przypisanie klawisza z czwartej listy	Send keyup event instead of keydown
X axis	Wciśnięty	Raw key	Any	CC	Eval. Item Val > Constant	Left 0 Right 64	Cursor left	zaznaczony
X axis	Puszczony	Raw key	Any	CC	Eval. Item Val <= Constant	Left 0 Right 64	Cursor left	odznaczony
X axis	Wciśnięty	Raw key	Any	CC	Eval. Item Val < Constant	Left 0 Right 192	Cursor right	zaznaczony
X axis	Puszczony	Raw key	Any	CC	Eval. Item Val >= Constant	Left 0 Right 192	Cursor right	odznaczony
Y axis	Wciśnięty	Raw key	Any	CC	Eval. Item Val > Constant	Left 0 Right 64	Cursor up	zaznaczony
Y axis	Puszczony	Raw key	Any	CC	Eval. Item Val <= Constant	Left 0 Right 64	Cursor up	odznaczony
Y axis	Wciśnięty	Raw key	Any	CC	Eval. Item Val < Constant	Left 0 Right 192	Cursor down	zaznaczony
Y axis	Puszczony	Raw key	Any	CC	Eval. Item Val >= Constant	Left 0 Right 192	Cursor down	odznaczony

Tabela 1



x, na ostatnim przycisku fire kończąc. Ostatnia lista, po prawej na dole, pokazuje domyślne ustawienia dla poszczególnych przycisków z poprzedniej.

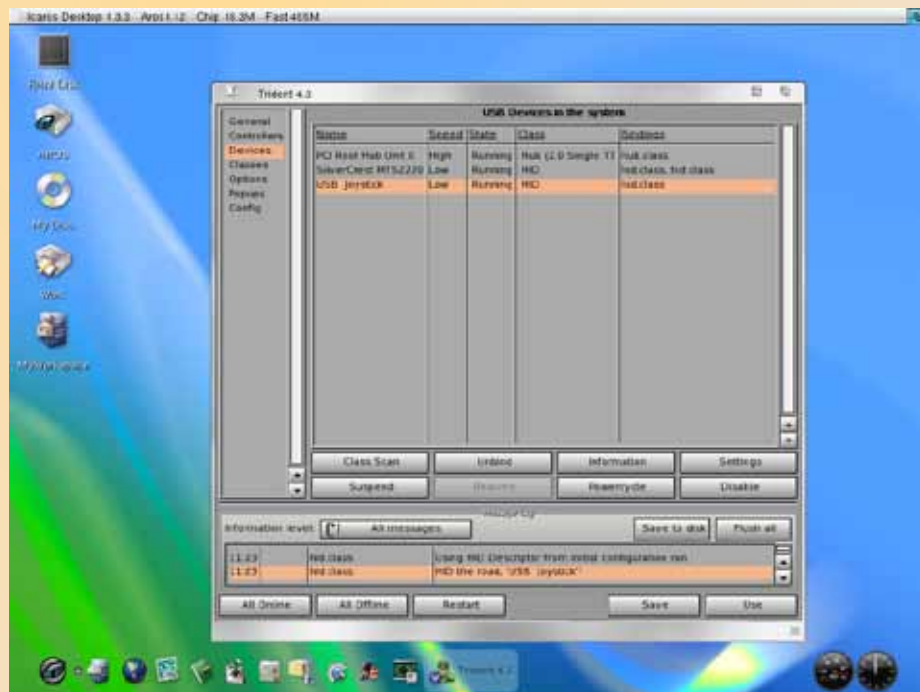
Jak widać wyraźnie, są to przypisania cyfrowe i analogowe. Tych ustawień, także nie będziemy zmieniać. Będziemy natomiast do każdego przycisku z listy drugiej przypisywać klawisz z klawiatury.

Zanim to zaczniemy, kilka uwag. Każdy klawisz przypisane będzie miał dwa polecenia; **klawisz wciśnięty** i **puszczony**. Nowe funkcje muszą znajdować się na końcu prawej dolnej listy, bo jeśli dana gra korzysta z *lowlevel.library*, w pierwszej kolejności będą brane pod uwagę, polecenia cyfrowe i analogowe, na końcu nasze. Funkcja **wciśnięty** i **puszczony** powinny być w tej właśnie kolejności (jeśli wpisałem na odwrót, pad czasami dziwnie się zachowywał). Po każdej nowo wklepanej funkcji, klikamy na **save** (lub **save as default** – zdarzało się, że gdy wprowadziłem wszystkie ustawienia, po zapisie część zniknęła, każdorazowe zapisywanie uchroni nas od powtarzania tejże żmudnej czynności, jedyne co będziemy musieli powtarzać, to otwieranie za każdym razem okna *hid.class*).

Z górnej lewej klikamy na pierwszą pozycję (**Joystick(Joystick)**), prawa górna lista – oś x (**X axis**). Następnie pod listą u dołu klikamy na **New**. Pojawi się nam **No action**. Teraz przechodzimy do prawej dolnej części. Klikamy na gadżet cykliczny również zatytułowany **No action** i wybieramy **Raw key**. W **Trigger** wybieramy **Any**, a w kolejnej linii zaznaczamy gadżet **CC**. Pojawią się nam nowe trzy gadżety cykliczne. W pierwszym wybieramy **Eval. Item Val**, w drugim znak **>**, a w trzecim **Constant** (czyli stały). Wklepujemy wartość **Left Constant „0”**, a wartość **Right Constant „64”**. Z czwartej listy (ostatniej, w prawym dole; która pojawiła się w trakcie kon-

figuracji) wybieramy **Cursor left**, zaznaczamy **send keyup event....** (zaraz pod czwartą listą) i zapisujemy.

Klikamy znowu na **Settings**, przechodzimy do **Action**, pierwsza pozycja z lewej górnej listy, w drugiej liście **x axis**, pod trzecią listą **New** (ma być na końcu, jeśli coś się pomieszało, należy je przesortować klikając na **Up** lub **Down**). Następnie znowu **Raw key**, **Any**, **CC**, **Eval. Item Val**, teraz wybieramy **<=** i ostaniamy **Constant**. Wartości **Left Constant**



„0” i **Right Constant „64”**. Ostaniamy listą to oczywiście **Cursor left**, natomiast **Send keyup event...** ma być w tym przypadku odznaczona. Zapisujemy. Właśnie przypisaliśmy lewy kursor klawiatury przyciskowi w lewo na joysticku.

Jak widać z powyższych czynności, owa konfiguracja zmierza do pewnego rodzaju schematu. Ukazane zostało rozmieszczenie opcji i gadżetów. Tak więc żeby się dalej za bardzo nie rozpisywać odsyłam Was do tabeli 1.

Teraz zostały nam jeszcze klawisze fire. W moim przypadku jest 10 zwykłych i 2 w gałkach (stąd 12 buttonów na liście). Powód skonfigurowania „tylko” dziesięciu, podałem na końcu artykułu.

Z górnej prawej listy wybieramy **Button 1(primary/trigger)**. Na dolnej prawej liście, tradycyjnie pojawi się nam już gotowe ustawienie dla *lowlevel.library* (**Digital joystick**), którego nie ruszamy. Tutaj także musimy ustawić dwie akcje; **klawisza wciśniętego** i **puszczonego**. Klikamy na **New**, w **Performed action** (pierwszy gadżet cykliczny) wybieramy **Raw key**, w **Trigger** wybieramy **Any**. Następnie z prawej dolnej listy, wzorem przygotowanego wcześniej planu wybieramy klawisz z klawiatury, czyli w tym przypadku **Control**. Ptaszkujemy **Send keyup event ...** I zapisujemy. Powtarzamy całą operację, czyli znowu **Button 1**, **New**, **Raw key**, w **Trigger** tym razem wybieramy **Down**, klawisz **Control** na dolnej prawej liście, **Send keyup event ...** tutaj ma być odznaczona i zapisujemy. Pierwszy fire na naszym padzie właśnie został przypisany do klawisza **Ctrl** na klawiaturze.

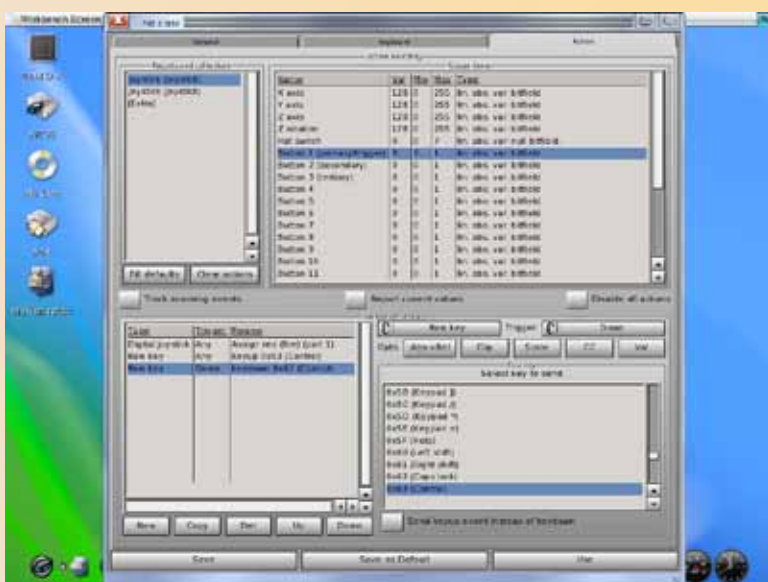
Jak zauważyliście, jedyne różnice w stosunku do klawiszy ruchu, to brak zaznaczonego gadżetu **CC** w linii **Opts** (a co za tym

idzie, niewymagane wprowadzanie wartości **Left** i **Right Constant**), oraz opcja **Down (Trigger)** w poleceniu klawisza puszczonego (zamiast **Any**).

Pozostałe klawisze fire konfiguruje się identycznie. Dla poszczególnych buttonów

Klawisz joja	Opcja klawisza	Performed actions	Trigger	Przypisanie klawisza z czwartej listy	Send keyup event instead of keydown
Button 1	Wciśnięty	Raw key	Any	Control	zaznaczony
Button 1	Puszczony	Raw key	Down	Control	odznaczony
Button 2	Wciśnięty	Raw key	Any	prawy Shift	zaznaczony
Button 2	Puszczony	Raw key	Down	prawy Shift	odznaczony
Button 3	Wciśnięty	Raw key	Any	Spacja	zaznaczony
Button 3	Puszczony	Raw key	Down	Spacja	odznaczony
Button 4	Wciśnięty	Raw key	Any	Return (Enter na głównej)	zaznaczony
Button 4	Puszczony	Raw key	Down	Return (Enter na głównej)	odznaczony
Button 5	Wciśnięty	Raw key	Any	Lewa Amiga	zaznaczony
Button 5	Puszczony	Raw key	Down	Lewa Amiga	odznaczony
Button 6	Wciśnięty	Raw key	Any	Prawa Amiga	zaznaczony
Button 6	Puszczony	Raw key	Down	Prawa Amiga	odznaczony
Button 7	Wciśnięty	Raw key	Any	Prawy Alt	zaznaczony
Button 7	Puszczony	Raw key	Down	Prawy Alt	odznaczony
Button 8	Wciśnięty	Raw key	Any	Lewy Alt	zaznaczony
Button 8	Puszczony	Raw key	Down	Lewy Alt	odznaczony
Button 9	Wciśnięty	Raw key	Any	Help	zaznaczony
Button 9	Puszczony	Raw key	Down	Help	odznaczony
Button 10	Wciśnięty	Raw key	Any	Delete	zaznaczony
Button 10	Puszczony	Raw key	Down	Delete	odznaczony

Tabela 2



naszych ustawień). Potem wystarczy zmajstrować dwa skrypty dla **IconXa**, który w zależności od potrzeby, będzie nam kopiował żądane ustawienia (najlepiej do **Env/Sys** w Ram Dysku, a np. w **EnvArc** zostawić jedną wersję ustawień i traktować ją jak domyślne – będą się ładować przy każdym starcie Arosa).

Tak czy siak, przedstawiony sposób konfiguracji joysticka w Arosie, działał bezproblemowo na przeważającej większości przetestowanych tytułów (biorąc pod uwagę akurat ten model pada). A dzięki zaplanowanej liście przypisanych klawiszy, w grach których jest to możliwe, mogłem ustawić własne preferencje.

Jeśli zechcecie przypisać klawiaturze więcej niż jeden kontroler, wpięty w komputer, sposób już znacie. Wystarczy tylko przygotować listę klawiszy dla drugiego joja.

Tradycyjnie, ze wszystkimi pytaniami i propozycjami (może komuś uda się udoskonalić mój projekt, wliczając w to bezproblemową obsługę galek) odsyłam Was do naszego redakcyjnego forum.

Don Rafito

zmienia się tylko przypisane klawisze klawiatury. Tradycyjnie tabelki z moimi przykładami.

Powyższa „recepta” powstała na zasadzie metody prób i błędów. Nie konfigurowałem klawiszy fire na galkach (11 i 12), gdyż uznałem że dziesięć to i tak wystarczająco, oraz dlatego iż nie udało mi się skutecznie przypisać samych ruchów galek dla padzie. Co prawda gałki były obsługiwane przez klawisze kursorów (po przełączeniu trybu na padzie), ale nie we wszystkich grach zachowywały się normalnie. Najlepiej spisywały się najzwyklejsze joje, bez żadnych udogodnień. Bardzo rzadko (dosłownie pojedyncze tytuły) zdarzało się, że preferencje **lowlevel.library** gryzły się z naszymi przypisaniami. I na to jest rada. **Posejdon** zapisuje swoje preferencje w **EnvArc/Sys** (dysk systemowy). Po zakończeniu konfiguracji, robimy sobie kopię tych ustawień, a następnie jeszcze raz przystępujemy do edycji, z tą różnicą, że we wszystkich klawiszach pada kasujemy polecenia cyfrowe i analogowe z dolnej prawej listy (zostawiamy same **Raw Key** – czyli te które sami wprowadziliśmy) i zapisujemy. Potem także robimy kopię tych ustawień (najlepiej np. zrobić sobie katalog „joy”, a w nim dwa osobne; „low” i „raw” i tam wrzucić kopie



Otej karcie turbo do Amigi 600 wiadomo już od ponad roku. Można także było przeczytać recenzję tej karty na stronie PPA. Teraz na łamach CAFan możecie, drodzy czytelnicy, przeczytać nie tyle recenzję lecz opis małych bojów z tą przystawką.

Z zakupem zbierałem się przez cały rok. Jak wiadomo czas nie stoi w miejscu i przez ten czas pojawiły się na internecie miłe i niemiłe informacje związane z tą przystawką do naszej pocztowej 600-ki.

Początkowo zamierzałem zakupić tą kartę w sklepie amigakit.com, jednak ostatecznie zakupiłem ją w ASB Computer znajdujący się w Hiszpani. Związane to było z faktem, iż w pierwszym sklepie spóźniłem się prawdopodobnie o tydzień z decyzją zakupu ;) Złożyłem zamówienie, wpłaciłem odpowiednią kwotę i należało czekać na samą przesyłkę. Otrzymałem ją półtora tygodnia później, co z uwagi na okres przedświąteczny jest zadowalającym wynikiem. Po rozpakowaniu paczki, ujrzałem upragnioną kartę, dwie śrubki służące do przykręcenia karty do płyty głównej Amigi oraz krótką ulotkę spełniającą rolę instrukcji. Sama ulotka odsyła nas na stronę producenta (www.icomp.de) w celu pobrania instrukcji w formacie *.PDF oraz programu ACATune, służącego do uaktywnienia pewnych dosyć ciekawych funkcji karty. Niestety, na stronie nie znalazłem linka do tychże plików, a przejrzałem wersję angielską i niemiecką tej strony. Nie zrażony tym niepowodzeniem, napisałem maila korzystając z formularza znajdującego się na stronie, że pomimo usilnych starań nie znalazłem żadnych linków do plików wymienionych w ulotce i nie jestem w stanie w pełni wykorzystać możliwości drzemających w karcie ACA 630/25.

Po kilku dniach otrzymałem maila z przeprosinami za brak tychże linków na stronie, a w załączniku otrzymałem owe pliki oraz nową wersję beta programu ACATune, obsługującego także karty ACA 1230/42. Wtedy zaczęło się testowanie karty czyli chwile zadowolenia oraz lekkiej rozpacz.

Samo testowanie zaczęło się już po zamontowaniu karty w Amidze, jeszcze bez posiadania programu ACATune. Montaż karty, po przejrzeniu zdjęć oraz przeczytaniu „zeznań naocznych świadków” na różnych stronach internetowych związanych z tematem amigowskim, nie nastręcza żadnych problemów. Wraz z kartą otrzymujemy bowiem dwie śrubki, za pomocą których przykręcamy kartę do płyty głównej, po wcześniejszym dociśnię-

ciu karty. Trzeba uważać, żeby podczas dociśnięcia czegoś nie uszkodzić, a należy to robić do momentu usłyszenia lekkiego kliknięcia, które informuje nas o wskoczeniu podstawki na procesor MC68000 na płycie głównej A600. Osobiście brakowało mi dwóch dodatkowych śrubek obok dwóch dołączonych do zestawu, dzięki którym mogłbym przykręcić radiator do płyty głównej, ale to nie przeszkadza w

ACATune, tylko z podstawowymi ustawieniami karty, jednak było co podziwiać. Karta jest bowiem skonstruowana w taki sposób, że nie potrzeba uruchamiać programu ACATune, aby móc wykorzystać jej podstawowe możliwości, czyli MC68030 i 32MB Fast RAM. Sama pamięć Fast przedstawia się jako pamięć zamontowana w slot Zorro III, co było dosyć ciekawym zaskoczeniem.

POTURBOWANA SZEŚĆSETKA

czyli jak rzecz ma się z ACA 630/25

użytkowaniu, ponieważ i tak karta dzięki przykręceniu nie ma możliwości „odskoczenia” z procesora MC68000.

Po zamontowaniu karty, zamknąłem obudowę, podłączyłem wszystkie wymagane przewody, a następnie włączyłem zasilanie. Od razu dało się odczuć przyspieszenie działania Amigi. Start od momentu włączenia, do momentu całkowitego wczytania WorkBench był na tyle krótki, że nie zdążyłem jeszcze zareagować a już mogłem robić cokolwiek. Różnica w porównaniu do gołej A600 jest od razu zauważalna, jednak uruchomiłem wszystkim dobrze znany program SysInfo i z lekką ekstazą przyglądałem się wynikowi testu wyświetlanym na ekranie. Co prawda nie był to test z użyciem

Po lekkim przyzwyczajeniu się do szybszego działania WorkBench, postanowiłem uruchomić kilka gier, które na podstawowej konfiguracji nie zachęcają do siedzenia przy komputerze. Sprawdziłem tytuły takie jak:

- Space Shuttle
- Gunship 2000
- Elite II: Frontier
- Cytadela
- Wing Commander

Wszystkie dostały tzw. „kopa” i zaczęły przedstawiać się bardzo zachęcająco, zmuszając mnie do dłuższego posiedzenia przy każdym tytule. Krótkie chwile spędzone podczas obcowania z wymienionymi tytułami pozwoliły mi nareszcie docenić te gry po raz ponowny :)



Szkoda tylko, że takiej karty nie posiadałem lata wstecz...

Jednak nadszedł czas na przetestowanie dwóch funkcji programu ACATune, które interesowały mnie najbardziej. Są nimi „maprom” oraz „cache”. Pierwsza z nich pozwala na zrobienie obrazu Kickstartu z kości ROM w pamięci Fast lub wczytanie takiego obrazu z pliku na dysku. Druga natomiast włącza cache procesora. Już po przerzuceniu ROMu do RAMu, Amiga dostała jeszcze większego przyspieszenia. Teraz Workbench biegł jak szalony. Po włączeniu drugiej funkcji, Workbench zaczął wręcz fruwać, jednak utraciłem pewną kompatybilność ze starszymi programami, które używałem wcześniej. Po prostu Amiga zawieszała się podczas uruchamiania takiego programu przy włączonym cache procesora.

Natrafiłem jednak na pewien problem z wczytywaniem pliku ROM z dysku. Początkowo po wczytaniu jakiegokolwiek ROM'u miałem czerwony ekran informujący o błędzie w ROM lub inne „wynałazki”. Nie wiedząc, co może być tego przyczyną, zrezygnowany nieumożliwością sprawdzenia OS3.9 na A600 zasięgnąłem języka na AmiBay'u. Rozwiązanie tego problemu okazało się dość trywialne, jednak trzeba o tym wiedzieć. Po prostu, w programie HD Toolbox w zaawansowanych ustawieniach partycji, należy zmienić wartość „Max.Transfer” na 0x1fe00. Standardowo jest ona ustawiana na 0xfffff co automatycznie powoduje błędy przy wczytywaniu większych plików.

Po stoczonej walce z parametrem Max.Transfer, mogę spokojnie wczytywać jakikolwiek ROM do RAM. Od razu zabrałem się do zainstalowania OS3.9 i sprawdzenia jak sobie radzi na Amidze. Sama instalacja przebiegała bez problemów, więc od razu zresetowałem Amigę w celu sprawdzenia szybkości dzia-

łania, a ta wg. niektórych źródeł nie zachęca. Sam się o tym przekonałem dosyć szybko. Wczytanie samej tapety zajęło Amidze ok. minuty po wczytaniu systemu. Odświeżanie zawartości okien i samych wyświetlań woła o pomstę. Je-

dynie kolorystyka w 16 kolorach jeszcze jakoś łagodzi te doznania, w niższej liczbie kolorów nie warto uruchamiać tego systemu. Po pół godzinnej męczarni z OS3.9 postanowiłem wrócić do OS3.1, połączyć go i podmienić nieco elementów z 3.9.

Po dłuższym czasie wybrałem na stałe 4 funkcje programu ACATune, które są wpisane w Startup-Sequence. Musiałem podzielić tą linię na dwie części, ponieważ jedna okazała się za długa i dwie ostatnie funkcje nie działały. Uruchamiam go na samym początku, jeszcze przed Setpatch. Wygląda to następująco:

```
ACATune -maprom ścieżka_dostępu_do_pliku -fastchip on >NIL:
ACATune -cache on -burst on >NIL:
```

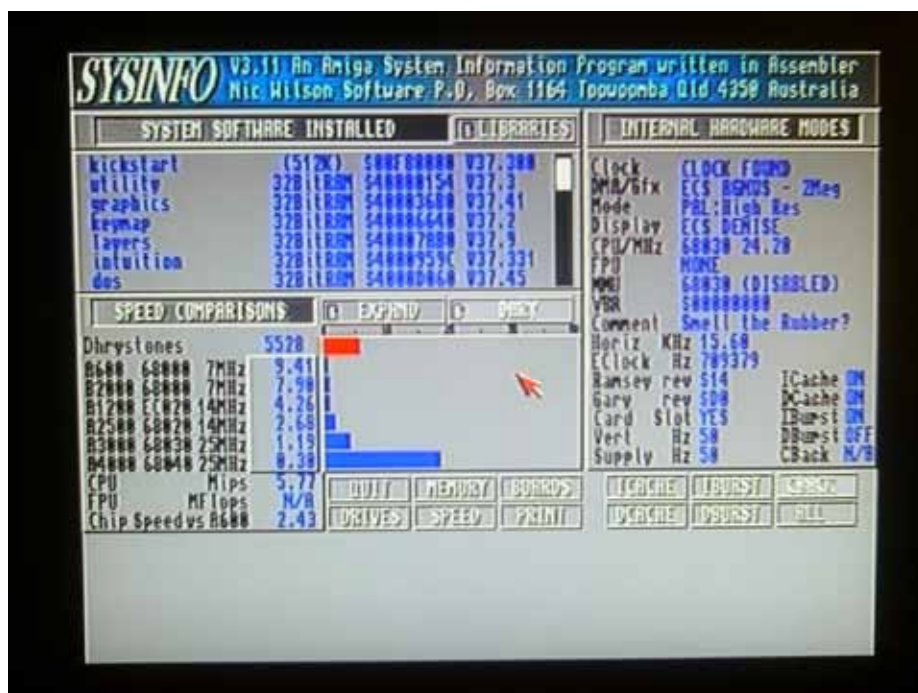
Co prawda można jeszcze włączyć cacheowanie pamięci Chip, przyspieszyć PCMCIA i użyć jeszcze kilku innych ciekawych funkcji, jednak może to powodować pewne problemy



z niektórymi programami. Dlatego postanowiłem zbytnio nie eksperymentować. Te funkcje, które aktualnie używam wystarczają w zupełności.

Zadać można sobie pytanie czy warto kupować tą kartę? Czynnikiem będącym przeciwko zakupowi tej karty są cena, „tylko MC68030 25MHz”, „tylko 32MB RAM”, pewne problemy z funkcjami programu ACATune. Natomiast za zakupem, przyspieszenie A600, rozszerzenie o dodatkowe 32MB RAM, efektywny montaż. Odpowiedź nie może być jednoznaczna, jednak z uwagi na niedostępność innych powstałych kart turbo, jest to bardzo atrakcyjna oferta firmy Individual Computers.

Wybór jednak należy do użytkownika Amigi 600. Musi on sobie postawić pytanie, czy warto dla niego poświęcić niemałą sumkę na turbo do Amigi.



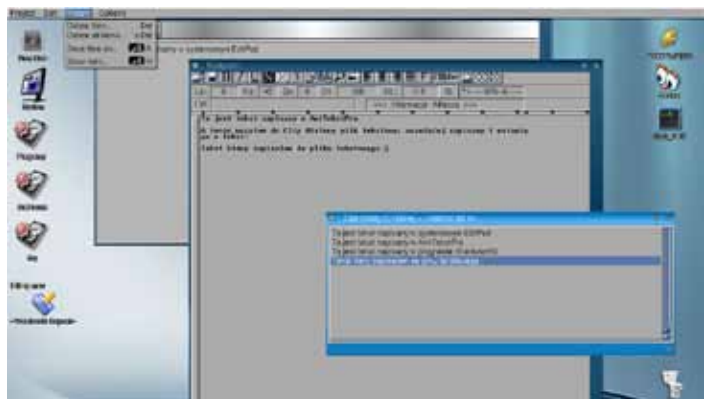
Aktualnie dostępne opcje to:

- Apollo 620 - turbo MC68020/20-25MHz + 0-8MB Fast
- Apollo 630/Winner 630 - turbo MC68030/33-50MHz + 0-32MB Fast
- M-Tec 630/Viper 630 turbo MC68030/40MHz + 4-8MB Fast
- ACA 630/25 - turbo MC68030/25MHz + 32MB Fast
- ACA630/30 - turbo MC68030/30MHz + 64MB Fast
- A608 - rozszerzenie Fast
- A6095 - rozszerzenie Fast i Slow
- Furia 628 - turbo MC68000/28MHz + 4-8MB Fast

Co prawda, część z nich jest ciężko dostępna, jednak trzy ostatnie są domowymi projektami, które mogą zaspokoić potrzeby niektórych osób.

Atreus

Clip History



Witam. Dziś przedstawię programik, a właściwie pchelkę, która bardzo ułatwi życie wszystkim tekściarzom.

Zdarza się tak, że podczas pisania tekstu (może nawet artykułu do CA Fan), używa się kilkakrotnie tych samych zwrotów. Zamiast jeździć za każdym razem po całym tekście w poszukiwaniu potrzebnego wyrażenia wystarczy uruchomić sobie programik *Clip History*.

Program jest typu commodity. Po pierwszym uruchomieniu siedzi sobie w systemie, nie pożerając wielkich zasobów, a jego zadaniem jest przejęcie kontroli nad systemowym schowkiem.



Od tej pory podczas pisania, możemy wczytać wiele różnych wycinków, które zostaną w pamięci komputera i będą do dyspozycji piszącego w każdym momencie.

Po wywołaniu okna Clip History, pojawi się lista skopiowanych fragmentów, wystarczy wybrać żądany dwukrotnym kliknięciem lewego klawisza myszy, a wycinek jest już dostępny w oknie edytora tekstu pod kolumną „wstaw” (lub jak kto woli „RAMiga” + „V”).

ny zapisany na dysku tekst można wczytać do clipboardu i użyć go później w edytorze. Można także kasować cały schowek, lub pojedyncze wycinki.

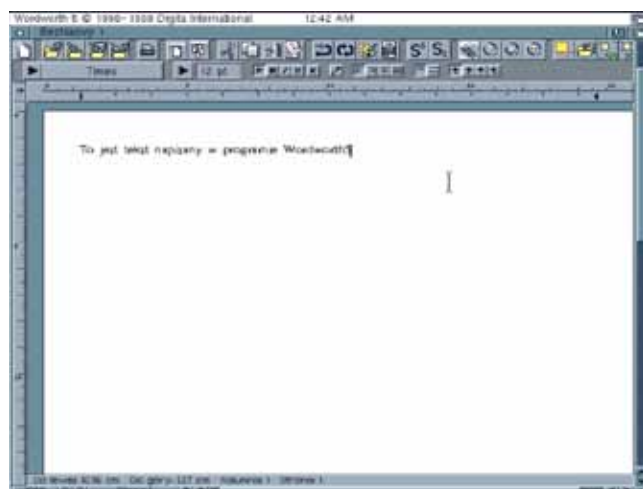
Konfiguracja Clip History sprowadza się do edycji tooltypes ikonki. Możemy między innymi zmienić klawisze skrótu (domyślnie „Ctrl” + „Alt” + „C”) wywołujące okno programu, przypisać skróty do przemieszczania się między poszczególnymi wycinkami (bez wywoływania okna programu), czy też ustawić, czy okno programu ma się pojawiać przy starcie. Możemy także ustawić priorytet, wskazać skrypt ARexxa, zdefiniować rozmiar i pozycję okna, czy też ustawić ekran publiczny dla programu.

Instalacja aplikacji sprowadza się do wrzuce-

Ale to nie koniec jego możliwości. Prócz przechowywania wielu różnych wycinków, program potrafi zapisywać je do plików tekstowych, jak i na odwrót; dowol-

która uzupełnia luki w amigowym systemie. Ale jak już kiedyś pisałem, takiej możliwości nie ma za wiele systemów, a przynajmniej nie w takiej skali, jak AmigaOS. A jedyne ograniczenia, to zasoby komputera i wyobraźnia użytkownika.

Don Rafito



Gatunek: użytek/WB

Producent: Magnus Holmgren (cmh@lls.se)

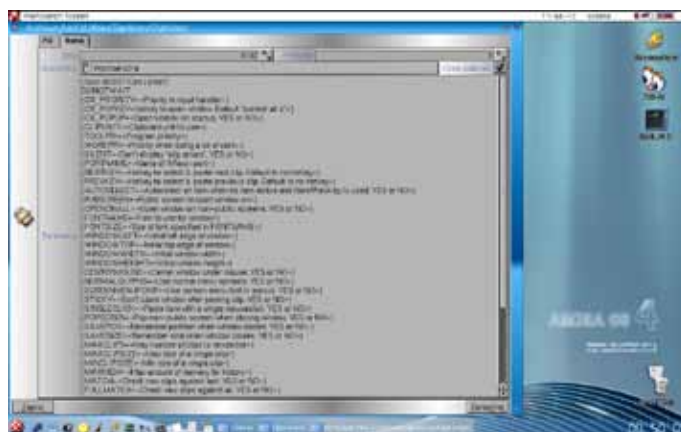
Dystrybucja: Amikit, Aminet

Rok: 1999

Rozmiar: lha 130 KB, HD

Platforma: mk68 (minimum OS 2.04)

Ocena: 90%



nia pliku wykonywalnego (wraz z ikonką) do WBstarup, lub do-pisanie ścieżki dostępu w user-startup.

Co tu więcej pisać, kolejna, godna polecenia, przydatna pchelka wchodząca w skład pakietu *Amikit*,

Amiga w kieszeni, czyli Uae4all na symbian



Przyznam się szczerze, że ten artykuł powstawał w otoczeniu jakiejś kłątwy. Przez ostatnie miesiące kilkakrotnie mijałem się z okazją wejścia w posiadanie smartfona opartego o system Symbian.

Kiedy parę tygodni temu udało mi się nabyć takowy sprzęt, uległ on nie wielkiej awarii. Po naprawie w końcu przystąpiłem do testów opisywanej aplikacji i zacząłem tworzyć artykuł. Ale tu znowu we znaki dały się siły tajemne. Tym razem komputer zaniemógł, a konkretniej partycja na której znajdował się prawie ukończony tekst. Partycja oczywiście uszła cało, ale wszelkie pliki już nie za bardzo.

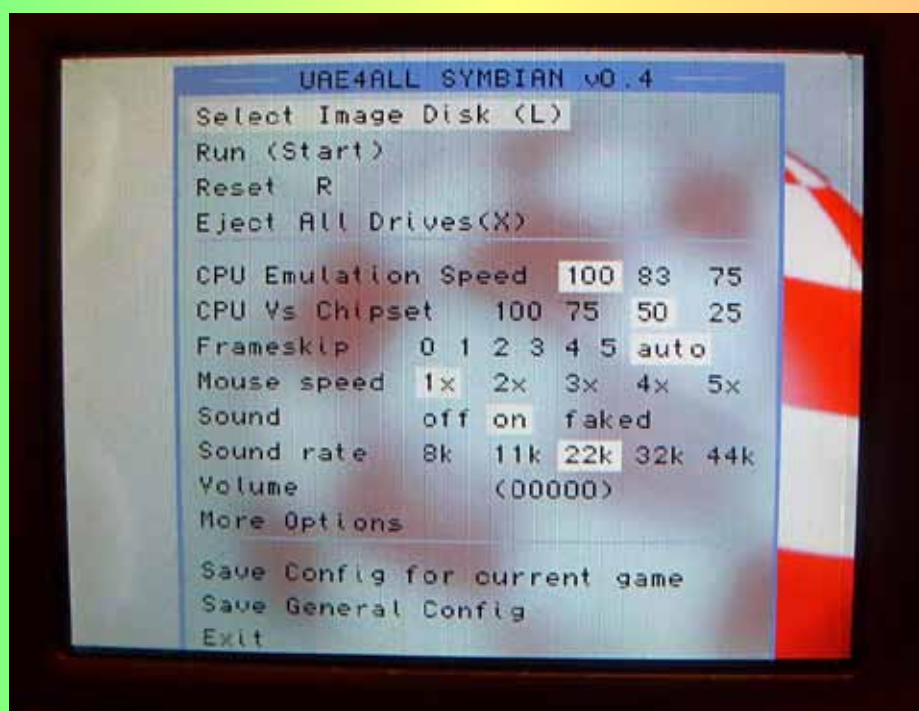
No nic, koniec końców wzbogaćony wiedzą na temat emulatora, a także doświadczony szeregiem zbiegów okoliczności, oto jestem i piszę ten głupkowaty wstęp do, mam nadzieję, ciekawego artykułu.

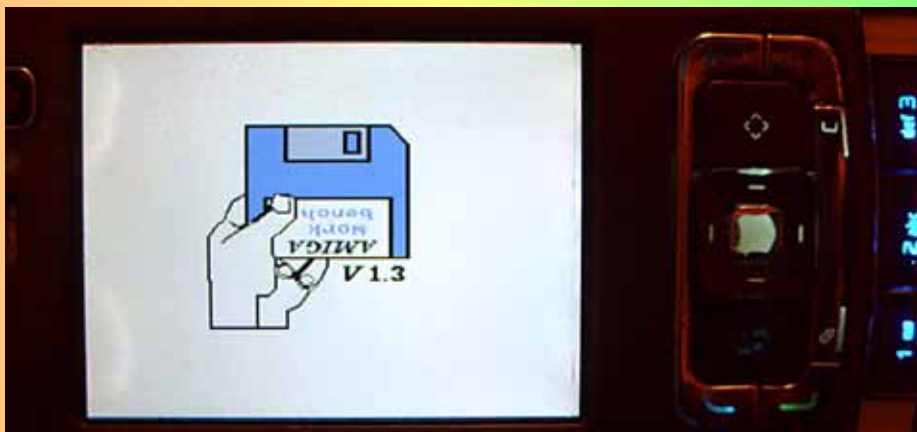
Przez ostatnie lata słowo „mobilny” jest coraz bardziej na topie. Sprzęt elektroniczny, technologie informatyczne, usługi telekomunikacyjne. Wszystko ukierunkowane jest na coraz większą dostępność w każdym miej-

scu i o każdej porze. I tak mniej więcej moż-

na właśnie przedstawić znaczenie tego słowa, które po przetłumaczeniu znaczy tyle co „przenośny, przewoźny, ruchliwy”. Internet, telefon, tablet, już nie dziwi nikogo widok laptopa na kolanach w parku na ławce. Niestety, spotykam się coraz częściej z przypadkami nadużyć znaczenia słowa „mobilny” w celach marketingowych. Mobilna kosmetyczka, mobilna myjnia samochodowa, mobilny konsultant, mobilny salon samochodowy (przenośny???), mobilne usługi (ciekawe jakie, chyba ruchliwe). To zaledwie schyłek góry lodowej tego co pokazało mi Google. No jeśli tak się sprawy mają, to w zasadzie codziennie mamy do czynienia z mobilnymi rzeczami. Np. bielizna, przecież mając ją na sobie (albo nie) zabieramy ją wszędzie ze sobą. Strach pomyśleć, co ci producenci wymyślą, jeśli „mobilny” wyjdzie z mody. Może „portable”. Widział ktoś kosmetyczkę portable? Może być ciekawie. Proponuję zatem tę całą mobilność zostawić elektronicznym gadżetom, a reszta wystarczy jak będzie przenośna.

A zatem, czy Amiga też może być mo-





bilna i na dodatek nie zabierać więcej miejsca niż paczka papierosów? Ależ może. Wszystko dzięki urządzeniom, takim jak telefon lub bardziej smartfon, opatrzone w odpowiedni system i hardware, oraz odpowiednie oprogramowanie.

T y t u ł o w y Uae4All, to nic innego jak sdl-owski port znanego nam Unix Amiga Emulator. Stworzony przez Another Guest'a na zasadach GNU Public License, dedykowany jest on dla telefonów opartych o system Symbian S60v3.

Jego najważniejsze cechy to; emulacja kości OCS, obsługa do czterech stacji dyskie-tek (pliki ADF), pamięć do 1 MB (tylko Chip), emulacja dźwięku, wirtualna klawiatura, mysz i joystick, a od wersji 0.4 obsługa kickstartów 1.2, 1.3, 2.0 i 3.1. W tej-że wersji możemy także przypisać dowolne klawisze amigowej klawia-

tury, przyciskom w telefonie i wykorzystać je potem w trybie klawiatura + joystick.

Reasumując, spokojnie możemy odpalić na telefonie Amigę 500 1 MB Chip. I choć używanie Workbench'a w tych warunkach nie będzie najwygodniejsze, to już odpa-



lanie najróżniejszych gier, przeznaczonych na



tę maszynę będzie dużo prostsze i możliwe do zrealizowania.

Może kilka słów o samej instalacji i konfiguracji. Pliczek (najczęściej) uae4all.sisx instalujemy w pamięci telefonu (można też na karcie). Następnie do katalogu c:\data\uae4all\kickstarts kopiujemy pliki rom, zachowując następującą zasadę nazewnictwa:

Kickstart 1.2 - kick12.rom

Kickstart 1.3 - kick13.rom

Kickstart 2.0 - kick20.rom

Kickstart 3.1 - kick31.rom

Jeśli jesteśmy posiadaczami Amigi to sprawa jest rozwiązana, wystarczy odpalić Transrom i zgrać kickstart do pliku. Jest on chroniony prawem autorskim, więc jeśli nie mamy takiej możliwości musimy go nabyć. I tu pojawia się problem. Pliki te oferowane przez firmę Clonato nie były rozpoznawane przez emulator. W sieci oczywiście są wersje które współpracują z aplikacją, ale ściągając je złamiemy prawo.

Ale jeśli kupimy gdzieś na aukcji internetowej kostkę z oryginalnym kickstartem, to w tym momencie staliśmy się posiadaczami legalnej kopii i spokojnie taki plik możemy już pobrać.

Po odpaleniu ukazuje się główne menu programu. W górnej jego części wybieramy plik ADF, startujemy i resetujemy Amigę, oraz odłączamy wszystkie podpięte dyskiety. W dalszej części możemy pobawić się ustawieniami prędkości emulacji oraz obsługą dźwięku. Pod More Options, kryją się dodatkowe funkcje. Najważniejsze to zmiana trybu PAL/NTSC, ilość pamięci i dyskie-tek, wersja kickstartu. Tu możemy sobie także włączyć autofire, czy też wspomnianą wyżej opcję mapowania klawiszy. Po powrocie do głównego menu na samym dole mamy dwie ważne funkcje, czyli zapis ustawień i zapis ustawień dla konkretnej gry. Ta druga przyda się na pewno wtedy, gdy prócz joysticka mamy przypisane konkretne klawisze dla danego tytułu. Ważna

informacja odnośnie dyskietek. W zależności ile ustawimy stacji, przypisujemy pliki obrazów do konkretnej przez wciśnięcie klawisza od 1 do 4 przy wyborze pliku ADF.

Do testów użyłem trzech spośród najbardziej znanych gier z poczwiej 500-tki, czyli Super Frog, Lotus The Ultimate Challenge oraz Pinball Fantasies. Gry prezentowały się dość znośnie, choć najlepszy efekt z użyciem dźwięku był w przypadku ostatniej pozycji. Zarówno Lotus, jak i Super Frog deczko szarpały. Dodatkowo podczas grania naszą super żabką występowały lekkie przekłamanie w grafice. Postać chwilami zniknęła, a kolory ekranu nie były poprawnie wyświetlane. Co do Pinballa sprawdził się tutaj system mapowania klawiatury.

Tak czy owak grać się dało i myślę że z czasem dobiore sobie listę gier, które bez większych problemów odpalą się i będą grywalne akurat na moim modelu telefonu, gdyż prędkość emulacji według autora portu w dużej mierze zależy właśnie od posiadanego smartfona.

Tak czy owak aplikacja jest warta polecenia i naprawdę możecie wierzyć lub nie, ale możliwość zagrania sobie w swoje ulubione pozycje podczas czekania w długiej kolejce w urzędzie skarbowym to fantastyczna sprawa. Program jest darmowy, na dodatek podpisany certyfikatem i naprawdę użyteczny.



Co do oceny, to raczej się tu nie wypowiem, gdyż uzależnione by to było od samego hardware'u a nie od aplikacji. Na koniec dodam, że Uae4All dostępny jest także dla komórek z systemem Android.

Don Rafito



Gatunek: aplikacja na telefon

Producent i dystrybucja: Another Guest (www.anotherguest.se)

Dystrybucja: Amikit, Aminet

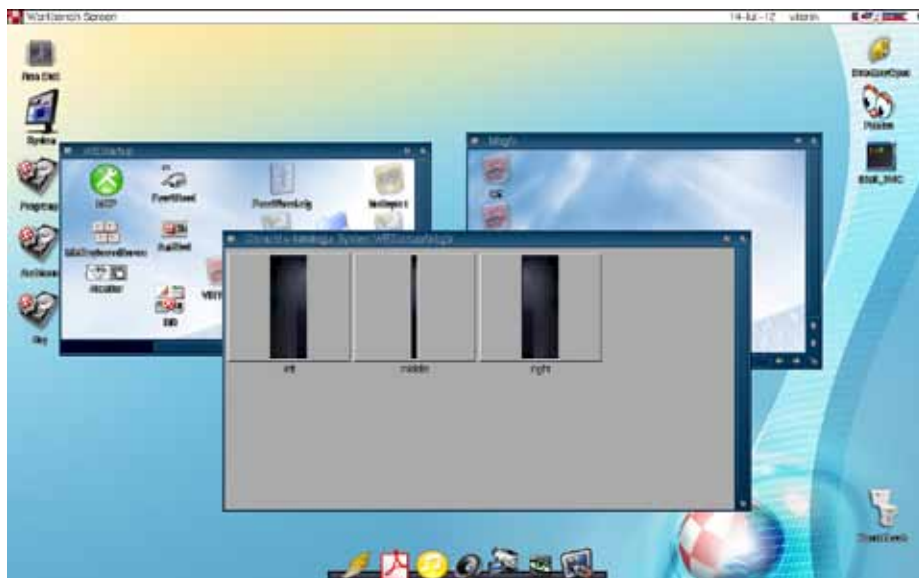
Rozmiar: Rozmiar 679 KB

Platforma: Symbian S60v3

LIPIEC 2012

Program do ustawień uruchamiamy na dwa sposoby; klikając na Bibprefs, albo wybierając „settings” z prawego menu wywołanego prawym klawiszem myszy po ukazaniu się belki. Prócz przypisywania samych programów, możemy je także grupować. Grupy z kolei będą dostępne w Boing Icon Bar pod wyżej wspomnianym prawym menu lub klawiszami funkcyjnymi (F2 i F3). Nowością przede wszystkim dla Amigi klasycznej, jest też to że po ukazaniu się paska, grupy można także przewijać rolką myszki (pod warunkiem, że mamy programik w systemie obsługujący rol-





kę, np. FreeWheel). Samą aplikację konfigurujemy poprzez edycję tooltypes ikonki. Można m. in. zmienić ścieżkę dostępu dla obrazka belki (wtedy katalog bibgfx nie będzie już potrzebny w WBstartup), rozmiar ikonki (MOS) oraz odstępy między nimi, a także wyłączyć auto-chowanie się.

Jeśli znudzi nam się standardowa bel-

ka, możemy ją zmienić, zastępując pliki graficzne o podobnych wymiarach w katalogu bibgfx.

Programik jest naprawdę godny polecenia, dla wszystkich, którzy cenią sobie ładny i schludny blat, a dotychczasowe rozwiązania deczko się znudziły. Cóż więcej dodać, niezła aplikacja, dobra dokumentacja, a przede

wszystkim pomysły i dostępność na wszystkie amigowe systemy.

Co prawda zabrakło tu opcji „przeciągnij i upuść”, ale myślę że autorzy dodadzą ją w kolejnych wersjach.

Don Rafito

Gatunek: użytek/WB

Producenci: Robert 'Phibrizzo' Krajcarz (jgn@interia.pl), Adam 'Adater' Terlecki, Paweł 'Stefkos' Stefański, Łukasz 'Sir_Lucas' Kucharuk

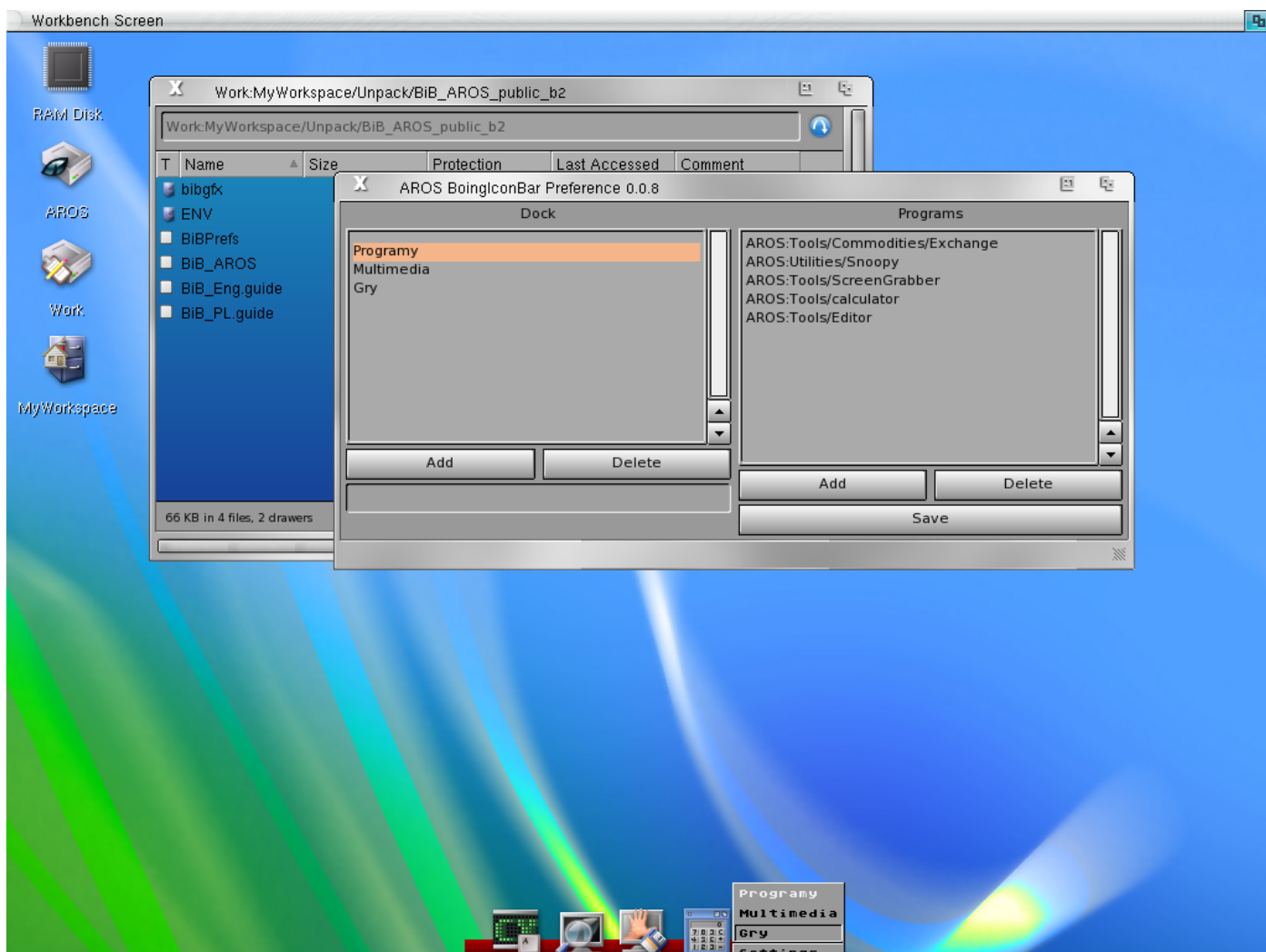
Dystrybucja: archives.aros-exec.org (wersja na Arosa), pozostałe wersje-kontakt z autorami

Rok: 2011/2012

Rozmiar: 1ha, 105 KB, HD

Platforma: mk68 (OS 3.5+), MOS 2.x, OS 4.x, Aros

Ocena: 90%



QUICK DATA DRIVE

W połowie lat 80-tych wprowadzony został na rynek czytnik danych produkcji Entrepo / Phonemark, przeznaczony do współpracy z komputerami VIC-20 i C64. Urządzenie to było znacznie szybszym rozwiązaniem niż standardowy magnetofon lub stacja dysków 1541, nazywało się Quick Data Drive i w chwili debiutu kosztowało 252 dolary.

Było to małe pudełko wielkości firmowego magnetofonu do C64, podpinane zresztą do jego portu, a za nośniki służyły mu specjalne kasety zwane Microwaferami. Sam zamysł stworzenia takiego urządzenia nie był nowością, pomysłodawcą i projektantem tychże nośników i urządzenia do ich odtwarzania była firma Sinclair Spectrum, która stworzyła pierwszą stację do ich odczytu i zapisu zwaną Rotronics Wafadrive w 1984 roku (które to z kolei było następcą jeszcze wcześniejszych podobnych konstrukcji). Niedługo później pojawił się właśnie Quick Data Drive.

Microwafer, czyli wykorzystywany przez urządzenie nośnik, na pierwszy rzut oka przypomina kształtem zwykłą, nieco mniejszą od standardowej, kasetę magnetofonową. Zawiera ruchomą osłonkę, która automatycznie usuwa się po włożeniu do napędu odsłaniając właściwy magnetyczny nośnik, posiada też przesuwane zabezpieczenie przed zapisem. W środku plastikowej obudowy znajduje się zapętlona taśma magnetyczna. Twórcy chwalili się, że cała „kasetka” jest dosyć trwała i nie musimy zbyt często drżeć z obawy przed jej przypadkowym uszkodzeniem, nie straszny będzie też dla niej choćby upadek. Należy ją jednak mimo wszystko chronić przed gorącem i wilgocią. Kasetki te występują w trzech rodzajach pojemności – od 16, przez 64 aż po 128kB. Maksymalnie są w stanie przechowywać 255 plików (które są umieszczane sekwencyjnie, jeden za drugim), jest też obsługiwany system katalogów.

Kampania reklamowa urządzenia była ukierunkowana na fakt uwypuklenia jego największej zalety – szybkości działania. Twórcy



chwalili się, że różnorakie przeprowadzone testy dowiodły, iż napęd ten jest blisko 20 razy szybszy niż standardowy magnetofon (oczywiście w trybie normal, gdyż w turbo ta różnica nie jest już tak porażająca) i 2 do 3-ech razy szybszy niż stacja dysków 1541. Powodem był fakt, że procedury przesyłania danych były dużo bardziej efektywne niż te od Commodore, a program przeznaczony do zarządzania plikami – bardziej współczesny i potężny. Tyle



reklamy, a jak przedstawia się użytkowa rzeczywistość?



Sam napęd Quick Data Drive podłącza się bezpośrednio do portu kasetowego w Commodore, niepotrzebne są żadne dodatkowe kable, interfejsy czy też zewnętrzne zasilanie. Urządzenie wizualnie wydaje się dość zgrabne, wymiary 118 x 50 x 148 mm są zbliżone raczej do magnetofonu niż stacji dyskietek. Na przodzie znajduje się dioda oznajmniająca momenty pracy silniczka napędu. Quick Data Drive może z powodzeniem pracować z naszym komputerem jako jedyny napęd, ale pozwala też na współpracę z Datasette (wpinamy je w cassette port w QDD, które staje się napędem przelotowym), lub drugim QDD (wpinamy analogicznie). Do zestawu dołączona jest kasetka Master QOS-Wafer, drugi pusty nośnik oraz instrukcja solidnie objaśniająca zarówno samo działanie urządzenia jak i wszelkie potrzebne do pracy z nim komendy. W momencie podłączenia i przystąpienia do pracy urządzenie początkowo emuluje standardowy magnetofon Datasette, by w trybie zwykłej prędkości wgrać



program umożliwiając podkręcenie tempa wgrzywania (quick load), a następnie załadować QOS oraz FMU.

QOS – Quick Operating System jest programem zawierającym zestaw niezbędnych komend służących do zarządzania plikami, podobnie jak ma to miejsce przy użyciu Datasette. Dodatkowo wybawia nas od konieczności manualnego wyszukiwania potrzebnych plików i używania przycisków play i record. FMU – File Management Utility udostępnia z kolei menu zawierające komendy służące do kopiowania plików pomiędzy QDD, stacją



dyskietek, magnetofonem, pozwala także na sformatowanie Microwafera (wymaganego przed pierwszym użyciem) oraz wyświetlenia listy znajdujących się na nim plików. Przy jego użyciu można także stworzyć QOS-Wafer (nośnik zawierający system QOS). FMU nie jest integralną częścią QOS, ponieważ jest zbyt duży, by był trzymany stale w pamięci. Jest wczytywany jedynie wtedy, gdy zachodzi taka potrzeba. Załadowany do pamięci Quick Operating System zajmuje komórki w zakresie od C000 do CFFF, więc należy być ostrożnym przy wczytywaniu programów również korzystających z tego obszaru, bowiem uniemożliwi to prawidłową pracę QOS. FMU z kolei, który rezyduje w miejscu pamięci RAM znajdującym się pod BASIC ROM, może zostać bezpiecznie nadpisany. Jak było wspomniane na początku, napęd współpracuje zarówno z C64 jak i VIC-20, jednak do jego działania wymagana jest odrębna wersja softu dla każdego z tych komputerów.

Istotną kwestią jest fakt, że QDD pozostaje praktycznie bezużyteczny bez Wafera zawierającego QOS (oryginału dołączanego do zestawu, bądź samodzielnie wykonanej kopii).



Po upływie lat okazało się niestety, że trwałość danych zawartych na Microwaferach pozostawia wiele do życzenia i zanikają one szybciej w porównaniu do standardowych nośników, co zauważają obecni posiadacze sprzętu. Tak więc ci, którzy chcieliby by ich niekonwencjonalny napęd był dziś czymś więcej niż jedynie muzealnym eksponatem i próbują zmusić go jeszcze do pracy – często gorączkowo poszukują możliwości nabycia sprawnego Wafera z QOS, bądź samego oprogramowania dającego możliwość utworzenia go. Natknąłem się również



na informację, jakoby czytnik w niektórych przypadkach, zamiast z QOS na Waferze, był zamiast niego sprzedawany z



kartridżem zawierającym tenże QOS – w tym wypadku jest gotowy do pracy od razu, bez wgrzywania niezbędnego oprogramowania do pamięci, no i sam kartridż jest na pewno sporo trwalszy. Wychodzi jednak na to, że to bardzo rzadka sprawa i egzemplarzy takich było naprawdę mało.

Tak naprawdę, jedyną chyba zaletą opisywanego napędu była szybkość jego działania. Jako, że czas odczytu i zapisu dochodził do wartości 2kB/sek., był to jak na tamte czasy parametr dosyć ciekawy. Urządzenie obsługiwało zarówno pliki programów, jak i dane sekwencyjne – te pierwsze zdecydowanie

szybciej, drugie mniej więcej z prędkością stacji dysków. Można więc powiedzieć, że docelowym przeznaczeniem sprzętu powinno być głównie częste wczytywanie dużych ilościowo programów czy gier. Tu jednak pojawia się problem, bo tak jak było wspomniane, obszar pamięci C000-CFFF musi pozostać nie naruszony, a wiele dużych gier i programów korzysta także z niego – konflikt ten oznacza nie lada kłopot.



Ponadto, zdecydowana większość softu na C64 i tak była sprzedawana na dyskach lub kasetach, więc siłą rzeczy użytkownik komputera był zmuszony do zakupu i użytkowania któregoś z dwóch napędów obsługujących te nośniki (lub obu oczywiście). QDD mógł być więc co najwyżej kolejnym dodatkiem i z tego też powodu jedynie bardzo znikomą część użytkowników Komcia rozważała jego zakup jako coś niezbędnego, czy chociaż potrzebnego. I choć cena urządzenia po pewnym czasie znacząco spadła i można było zaopatrzyć się w nie za kwotę 85\$ (w przypadku zakupu dwóch urządzeń jednocześnie, Entrepo oferowało dalsze rabaty, w sprzedaży były też korzystne cenowo zestawy z edytorem tekstu, organizatorami czy podobnym oprogramowaniem), to same nośniki w porównaniu do rozpowszechnionych już kaset czy dysków (dodatkowo bardziej pojemnych) pozostawały dość drogie.

Dziś, jak wiadomo, na tego typu peryferia patrzy się zupełnie inaczej, rzecz można, kolekcjonerskim okiem. Ulatują gdzieś wszelkie wady niezbyt udanego projektu, liczy się sam fakt posiadania czegoś nieszablonowego. A można powiedzieć, że urządzenie w zamyśle miało być czymś w rodzaju miksu stacji dyskietek i magnetofonu kasetowego. Jednak jak to określił jeden z recenzentów tego sprzętu, tak naprawdę używając go doświadczyliśmy „wszystkich minusów taśmy i żadnych zalet dysku”. Nic więc dziwnego, że sławy absolutnie nie zdobyło i po upływie czasu zostało po prostu zwyczajną ciekawostką.

noctropolis

Ewolucja dem



Oglądając demo, zazwyczaj nie zastanawiamy się, skąd się one wywodzą oraz jaką ewolucję przeszły. Aby mogły powstać dzisiejsze demo, najeżone wszelkiej maści spektakularnymi efektami, musiały przejść bardzo długą drogę od prostych dem typu intro do trackmo.

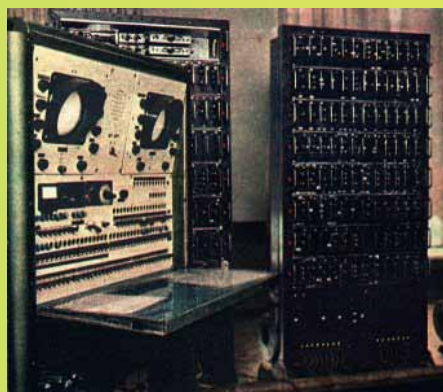
Czym więc jest demo? W dużym skrócie jest to program komputerowy, napisany najczęściej w assemblerze (jeśli chodzi o klasyczne komputery) lub w innym języku programowania dostępnym na danym komputerze. Celem jego napisania jest udowodnienie światu poziomu swoich możliwości w opanowaniu danej maszyny w różnych aspektach – programistycznym, graficznym i muzycznym. Chodzi o to, aby pokazać ile koder, grafik i muzyk jest w stanie „wycisnąć” z tego komputera. Demo najczęściej prezentowane są z dołączoną do nich grafiką i muzyką. Większość osób nie związanych ze środowiskiem demosceny może mieć wątpliwości po co takie demo robić i dla kogo? Tym bardziej, że jest to długi (niektóre demo powstają w pocie czoła nawet przez pół roku) i wyczerpujący proces. Okazuje się, że są na takie programy odbiorcy. Związani są oni głównie ze środowiskiem ludzi tworzących subkulturę demosceny. Sens jej istnienia polega na tworzeniu takich programów, aby pokazać innym jej członkom, że jest się od nich lepszym.

Na samym początku, twórcy dem chcieli pokazać projektantom komputerów domowych, że ograniczenia sprzętowe określone przez inżynierów da się łatwo obejść i wydobyć z tej maszyny znacznie więcej niż pierwotnie zakładali. Użytkownicy komputerów często udowadniaли inżynierom, że nie posiadając dużej wiedzy, można bez problemu pokonać bariery danego komputera. Tak było np. na C64, kiedy programiści pokazali, że da się otworzyć wszystkie boczne ramki, uzyskać więcej niż 16 kolorów czy wyświetlić na ekranie więcej niż 8 sprajtów.

Cofnijmy się do czasów, kiedy w domach nie było jeszcze komputerów, a można

je było spotkać tylko w dużych zakładach czy na Wyższych Uczelniach. Pierwsze komputery zajmowały nieraz całe pomieszczenia i nie miały zbyt dużej mocy obliczeniowej. Dla przykładu pierwszy elektroniczny komputer cyfrowy XYZ miał prędkość ok. 680 kHz, co daje 650-4500 dodawań na sekundę lub 350-500 mnożeń na sekundę, tak było w 1958 roku. Obecnie o wiele szybszy procesor mają najprostsze kalkulatory. W tamtym okresie pierwsi programiści, czasami z nudów, czasami z ciekawości, tworzyli pierwsze programy, których opis może pasować do obecnych programów demonstracyjnych.

To na nim w 1958 roku zaprezentowano pierwsze polskie demo. TVP1 wyemitowała reportaż o XYZ i zaprezentowano możliwości tego nowego wtedy komputera. Jak się okazało, telewizjom najbardziej przypadł do gustu zaprogramowany dla żartu bardzo uproszczony obrazek ze świecących punktów na oscyloskopie przedstawiający na ekranie pieska, który dodatkowo podnosił nogę i podlewał symboliczne drzewko. Można powiedzieć, że była to pierwsza animacja stworzona na komputerze w Polsce. Na początku była to tylko zabawa mająca na celu odkrywanie nowych możliwości pierwszych komputerów, które jak się szybko okazało mogły służyć do czegoś innego niż tylko do obliczeń matematycznych. Dobrym przykładem może być tu opowieść pana Jarosława Sokołowskiego, z którym przeprowadzi



Komputer XYZ z 1958 roku.

łem mały wywiad. W latach 70-tych XX wieku stworzył, jako jeden z pierwszych w Polsce, muzykę na komputer ODRA. Oto jak wspomina on tamte czasy: „Zacząłem programować (na Odrze w Fortranie) w połowie lat '70. Jako bardzo młody człowiek, licealista. To nie było wtedy popularne zajęcie licealistów, ale coś ja akurat mogłem. Wiedziałem więc, o co w tych całych komputerach chodzi. I to było fascynujące. Mogłem policzyć coś, co nie było do policzenia w inny sposób. Nie tylko policzyć, zrealizować algorytm, który wymagał co i rusz podejmowania decyzji. Dlatego zdawałem sobie sprawę, że komputer świetnie nadaje się na przykład do sterowania samochodem tak, by nie trafił w płot, nie wyrzucił się na zakręcie i



Polskie cracktro grupy Quartet

nie zderzył z innym pojazdem. Ale żeby stawiać wszystko na głowie, czyli że ja mam uważać na te płoty i krowy łażące po szosie, a komputer ma mi tylko sprawiać trudności (jak to w grach) -- co to, to nie....

„Tworzenie muzyki”, to określenie trochę na wyrost. Drukarka wierszowa podłączona do mainframe robi niezły łomot. Program, którego nie udało się bezbłędnie zrealizować, drukował tak zwany „trace”, czyli historię tego, co się działo zanim został zakończony. Wydruk „trace” jest dość nudny, a dźwięk drukarki był monotony. Wszyscy w okolicy „na ucho” potrafili rozpoznać, komu obliczenia się powiodły, a komu nie. No i nasłuchiwali. Skoro tak, to postanowiłem w ten sposób ustawić między wydrukami pętle z czasochłonnymi obliczeniami, by drukarka wystukała rytm Bolera Ravela.”

Tak mniej więcej przedstawiały się pierwsze próby tworzenia czegoś innego na dostępnych w kraju komputerach. Takich przykładów można by podać jeszcze kilka, ale jest to materiał na kolejny artykuł. Przejdźmy do lat 80-tych, kiedy to na dobre w domach zaczęły pojawiać się pierwsze komputery. Od tego okresu można mówić o rodzeniu się tzw. demosceny oraz tworzeniu przez nią pierwszych dem. Powszechnie uważa się, że pierwsze demo zostały stworzone przez crackerów, którzy łamiąc zabezpieczenia w grze często dołączali przed właściwą grą swoje programy zwane intrami, których celem było rozreklamowanie danego crackera. Była to także informacja komu pierwszemu udało się odebzpieczyć daną grę i wypuścić ją na giełdę. Po pewnym czasie te proste intra ewoluowały i tworzący je programiści przywiązywali większą uwagę do estetyki wykonania oraz coraz bardziej skomplikowanych efektów. Takie produkcje po pewnym czasie zaczęły funkcjonować jako odrębne dzieła, a ich autorzy zaczęli tworzyć subkulturę zwaną demosceną. Tak w dużym skrócie można opisać jedną z dróg, z której wywodzą się demo. Samo zaś zjawisko tworzenia tych programów sięga znacznie wcześniejszych lat, jeszcze przed powstaniem sceny crackerskiej.

Najczęściej przysłowiowemu Kowalskiemu, kiedy znudziła się już zabawa w gry komputerowe, pragnął wtedy stworzenia czegoś na swoim komputerze na miarę jego „wielkich” możliwości. Na samym początku



Grafika w PETSCII z demo Seasons Greetings

najczęściej zaczynał bawić się BASIC-iem oraz przerabianiem za pomocą cartridge'ów cudzych programów. Tym sposobem poznawał komputer i jego możliwości. Bardziej zdolni ludzie zabierali się za assembler. Najczęściej wyciągali z ulubionych gier muzykę plus jakiś obrazek i łączyli wszystko w całość. Tak powstawały pierwsze programy demonstracyjne, które z dzisiejszej perspektywy określić możemy jako kolekcje muzycznej. Wtedy nazywało się to demem. Takich programów powstawała cała masa, bo ludzie chcieli posłuchać ulubionych muzyczek z gier, bez konieczności ich uruchamiania. Wśród tych wszystkich programów trzeba było się jakoś wyróżnić, więc koderzy wymyślali coraz to bardziej urozmaicone sposoby pokazywania grafiki oraz wstawiali

nowe efekty przyciągające uwagę odbiorcy, takie jak dziwaczne scrolle oraz animacje. Dzięki tym zabiegom, twórca liczył na szersze rozpoznanie swojej pracy. W tamtych czasach nie było Internetu i wszystkie programy kopioowało się w domu. Potem wysyłało się je drogą pocztową do swoich znajomych lub też zanośiło na giełdę komputerową, gdzie handlarze byli w stanie rozprowadzić daną produkcję wśród sporego grona zainteresowanych.

Przyjrzyjmy się dokładniej, jak to wszystko wyglądało w przypadku C64. Pierwsze programy demonstracyjne zostały stworzone przez ludzi związanych z firmą Commodore. Były one najczęściej napisane w BASIC-u, w celu pokazania podstawowych możliwości nowego komputera wchodzącego dopiero na rynek. W ten sposób w 1982 pojawiło się trochę utworów muzycznych napisanych w BASIC-u przez twórcę SID, Boba Yewchuka, a także Jima Butterfielda. Na ekranie pojawiał się tekst, a w tle leciała muzyka skomponowana w BASIC-u, najczęściej był to jakiś znany cover. Nie były one niczym rewelacyjnym, ale miały za zadanie pokazanie nowych możliwości muzycznych SIDA, pokazy znacząco odbiegały od dokonań muzyków działających na scenie w późniejszym okresie. Jeszcze w 1983 roku firma Commodore wydała demo reklamowe o nazwie Seasons Greetings. I od tego momentu można już mówić o pierwszym demie na tym sprzęcie. Oprócz kilku utworów ze znanych koled dodano także prostą animację w PETSCII. Kolejne jej części dogrywały się z dyskiety. Dopiero rok później zaczęły ukazywać się w większej ilości pierwsze prymitywne demo. Właściwie były to muzyka plus grafika połączone z jakimś efektem, najczęściej scrollem, lub tekstem na ekranie. Od 1985 roku można śmiało mówić już o dynamicznym rozwoju dem. Coraz częściej zaczęto dodawać obrazki oraz coraz bardziej rozbudowane i finezyjne scrolle. Od tego roku zauważalnie wzrosło zainteresowanie ludzi tworzeniem prostych programów demonstracyjnych i łączeniem się w grupy typowo demoscenowe.

Przełomowy okazał się rok 1986. Wtedy to w demach zaczynały pojawiać się pierwsze efekty „niemożliwe” lub trudne do wygenerowania według specyfikacji technicznej. Holenderska grupa 1001 Crew zaczęła wtedy tworzyć demo z efektami, które pokazywały coś, czego inżynierowie z Commodore nie przewidzieli. Zaczęła się zabawa w „otwieranie” bocznych i górnych ramek oraz umieszczanie na nich grafiki w postaci sprajtów czy zabawy w różne rodzaje scrolli. Jednak wciąż cała praca wykonywana była przez samego koderza. Muzykę i grafikę wyciągano jeszcze z gier. Tylko nieliczne grupy posiadały swojego muzyka. Z grafiką było nieco lepiej, zaczęli pojawiać się pierwsi graficy, którzy w różny sposób upiększali te demo.

Rok później, w 1987, w demach pojawiły się znacznie lepsze, udoskonalone efekty.

Były one oparte na cyklowaniu. Najważniejsze z nich to: splits, raster bary i rastry. Cyklowanie to takie „układanie” rozkazów w programie, aby uzyskać opóźnienie z dokładnością do pojedynczego cyklu procesora. Te efekty nie były w większości oparte na wzorach matematycznych, a jedynie na dobrym kombinowaniu. Choć czasami zdarzały się wyjątki. Koderzy pokazywali sztuczki programowe, które na innych komputerach 8-bitowych były nie do osiągnięcia. Skoro C64 mógł wyświetlić tylko 8 sprajtów, to należało pokazać niedowiarcom, że da się ich wyświetlić na ekranie znacznie więcej. Ktoś wyświetlił ich 9, kolejny 12 i tak pobijano rekordy. Zaczęła się rywalizacja który koder, a później która grupa jest najlepsza. Widać to dobrze po statystykach na stronie CSDB (<http://noname.c64.org/csdb/stats/stat.php>) kiedy i jacy ludzie na demoscenie byli najbardziej aktywni. Od 1987 do 1989 roku tworzone ponad 9 tysięcy produkcji rocznie. Od 1990 zaczęło systematycznie spadać zainteresowanie tworzeniem na C64. Wtedy okres tzw. złotych lat demosceny C64 definitywnie się skończył.

W Polsce trochę inaczej to wszystko wyglądało. Jeśli chodzi o komputery domowe mieliśmy spore opóźnienie, ale nie jest to tematem tego artykułu. Na początku polskie demo składały się z jednej części z jednym efektem, ale czasy szybko się zmieniały i zaczęto do jednoczesnej produkcji dodawać kolejne części, do których przechodziło się po naciśnięciu odpowiedniego klawisza na klawiaturze (najczęściej spacji). Wyglądało to tak jakby się oglądało kolekcję intr z cracków, tylko że z lepszymi efektami.

Długo królował w domach zestaw komputer plus magnetofon i stąd tak duża popularność produkcji jedno plikowych. Kiedy na dobre w domach zagościły stacje dysków, zmieniła się także jakość produkcji na demoscenie. Posiadanie stacji dysków było wygodne, a wygoda ta została wymuszona przez producentów gier i oprogramowania. Ludzie piszący demo zaczęli powoli wykorzystywać nowe narzędzie do swoich produkcji. Mogli teraz tworzyć demo dogrywane z dyskiety. Zazwyczaj po bieżącej części demo plikowego dogrywała się następna część zamieszczona w innym pliku. Ładowanie kolejnej części odbywało się najczęściej standardową procedurą load, a na ekranie nic się nie pokazywało. Z nowym narzędziem, jakim była stacja dysków 1541, ludzie musieli się oswoić i poznać, zanim zaczęli używać jej niepublikowanych możliwości w swoich produkcjach. Stacja dysków przyczyniła się do stworzenia pewnych standardów, bez których demo dyskowe dziś nie może już istnieć. Zresztą wśród kilku stacji dysków do C64 to 1541 II stała się standardem i na niej opiera się tworzenie produkcji scenowych. Demo plikowe doczytywane z dyskiety umarło śmiercią naturalną, a w jego miejsce pojawiło się demo dyskowe zwane megademem. Był to popularny pod koniec lat 90-tych sposób two-

zenia dem. Luźno ze sobą skojarzone części, łączono w jedną dużą produkcję pod szyldem jakiejś grupy. Oglądający mógł w danej chwili przyspieszać pojawianie się kolejnych części przez wciśnięcie odpowiednich klawiszy, najczęściej spacji. W początkach lat 90-tych jako mega demo często określano demo plikowe, które posiadało dużo części. Z czasem, kiedy ludzie na scenie bardziej poznali możliwości stacji dysków, zaczęli wykorzystywać je w swoich produkcjach. Wtedy na scenie narodziła się profesja kodera stacji dysków. W Polsce jednym z pierwszych koderów stacji dysków 1541 był Krzysztof Matuła (K.M.) z grupy Taboo. Większość koderów z naszego kraju, choć nie tylko, długo używało jego loaderów w swoich produkcjach. To za sprawą jego IRQ-loaderów, które wczytywały na przerwaniach. Dzięki temu można było wyświetlić w czasie ładowania kolejnej części dema jakiś obrazek i odegrać muzykę. Dodatkowo loadery te, były kilka razy szybsze od standardowego loadera zawartego w KERNAL-u w stacji dysków. Kiedy tylko pojawił się IRQ-loader ludzie zaczęli go doceniać i używać w swoich demach. Z czasem IRQ-loadery stały się standardem, bez których dyskowa produkcja scenowa nie mogła się obejść.

Duży wpływ na jakość dem miała Amigowa scena. To dzięki niej mamy takie produkcje jak dentro i trackmo. W czerwcu 1993 roku na Elysium & Antia Party po raz pierwszy polska grupa Elysium w demie Obornik pokazała nowy typ dema tzw. dentro. Jest to demo plikowe, w którym efekty są samoczynnie pokazywane jeden po drugim (bez konieczności wciskania spacji), a muzyka gra nieprzerwanie. Najczęściej jest to jeden długi utwór, rzadziej więcej. Swego czasu był to bardzo popularny typ dema w naszym kraju.



Dentro Obornik / Elysium

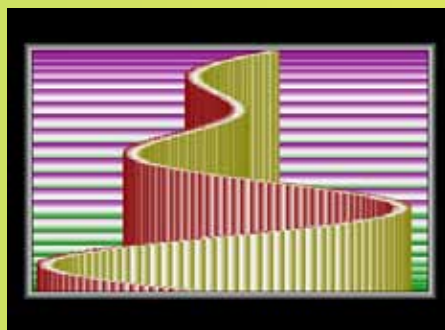
Drugim rodzajem dema przejętym ze sceny amigowej jest trackmo. Jest to całodyskowe demo, a raczej dentro, które czyta dane nie z plików, tylko bezpośrednio z track-ów (ścieżek) dyskiety. Poza tym ich cechą jest to, że następne efekty dogrywają się w trakcie (w tle) wyświetlania obecnych na ekranie. W takich demach główny nacisk kładzie się na design (po angielsku design).

Design polega na odpowiednim zgra-

niu efektów zaprogramowanych przez koderą oraz odpowiedniej do ich tempa muzyki w jedną całość. Nie chodzi tu tylko o synchronizację dźwięku z oglądanym obrazem, ale o podporządkowanie całości pewnej koncepcji, co do wyglądu dema jako całości. Dzięki temu zabiegowi dema zaczęły przypominać teledyski.

Na Amidze pierwsze trackmo „Mental Hagover” grupy Scoopex pojawiło się w 1990. Na C64 można mówić o pierwszych takich demach już od 1993 w produkcjach grupy Reflex z Niemiec. Wraz z kolejnymi produkcjami zmieniało się podejście do robienia efektów. Zamiast tworzenia efektów opartych na cyklowaniu, zaczęto tworzyć efekty liczone matematycznie przez procesor w czasie rzeczywistym (tzn. to co oglądamy na ekranie jest wynikiem aktualnie przeprowadzanych kalkulacji). Mówimy, że są one tworzone w „realtime”. Często musiały one przejść długą ewolucję, zanim ktoś wpadł na pomysł, że procesor C64 da radę wygenerować je w czasie rzeczywistym. Takim przykładem są efekty zwane wektorami. Najczęściej były to obracane w trzech lub dwóch wymiarach bryły geometryczne (sześciiany, ostrosłupy.. itp.) lub figury matematyczne (trójkąty, kwadraty... itp.). W pierwszych produkcjach były to zwykłe animacje, gdzie w demie były zawarte poszczególne jej klatki. Później zmieniono je na tzw. precalc, czyli w przeciwieństwie do animacji efekt częściowo wcześniej obliczany, którego wynik zawarty był w kodzie programu. Następnie przyszły „druciaki”, czyli wektory, w których ścianki nie były wypełniane kolorem. Oczywiście na samym początku był to precalc, a dopiero później zaczęto robić je w realtime. Zawsze w scroll-u można było przeczytać, że dany efekt jest w realtime, bo koderzy podkreślali dzięki temu swój kunszt. Potem przyszły wypełniane wektory w realtime, a wraz z nimi wektory TV-Box. Tak określano wektory, w których na ściankach były umieszczone jakieś proste efekty nierzadko inne wektory. Dużo efektów matematycznych pochodziło z Amigi, a pod koniec lat 90-tych także z PC.

W 1993 roku powstał między innymi program użytkowy 41 Track Format, które wykorzystywały dodatkowe ścieżki na dyskietce sformatowanej w stacji 1541. Normalnie stacja dysków formatuje do 35 ścieżki. Dzięki



Vertical Raster Bars z produkcji na Amigę Copper Master / Angels

specjalnym formaterem, można było sformatować dyskietkę aż do 42 ścieżki, uzyskując dodatkowo ok. 200 bloków. Koderzy używali formatowania tylko do 40 ścieżki, ponieważ dwie dodatkowe nie dawały się odczytać na wszystkich stacjach dysków. Wykorzystując specjalny loader, koder miał dodatkowe miejsce na dyskietce na swoje dane. Niestety na takie dodatkowe bloki powstały tylko cztery produkcje: Krestology/Crest, Digital World/Samar, Pearls for Pigs/Xenon oraz najnowsza



Krestology / Crest

Algotecher stworzona w całości przez koderą Algorithm. Zazwyczaj dodatkowe bloki wykorzystywali swaperzy, kiedy nie było miejsca na nagranie listu do swoich kontaktów.

Oprócz normalnych dem z efektami, istniały tzw. fake dema, a na Amidze nazywane jako crazy demo. Był to osobny gatunek produkcji scenowej, wydawanej najczęściej podczas party. Głównym ich celem jest rozbawienie oglądających przez różnego rodzaju gagi i aluzje do prawdziwych dem, rzeczy, sytuacji życiowych, osób itp. W większości takie produkcje raczej udawały prawdziwe demo, a jego autorzy nie dbali o maksymalne dopracowanie swojego dzieła zarówno od strony technicznej jak i artystycznej. Choć czasami zdarzały się wyjątki.

A jak wyglądają obecne dema? Nadal popularnym demem jest trackmo, w którym królują nowe efekty z dem pecetowskich takie jak twister lub stare efekty w nowych odsłonach takie jak vertical raster bars znany bardziej jako kefrens bars.

Zresztą wszystkie stare efekty znacząco się zmieniły i zostały dopracowane. Dzięki temu w cyklowanie prawie nikt już się nie bawi. Za to pojawiły się nowe kategorie produkcji scenowych, takie jak 4KB Intro czy efekt zajmujący 256 bajtów. Jest to krótkie trackmo, zazwyczaj nie zajmujące więcej niż 4KB. Istnieją też inne kategorie, jak demo 1KB. Wszystko zależy od ilości pamięci, jaką chcemy przeznaczyć na nasz produkt. Natomiast 256 bajtowe demo jest raczej pokazaniem przez koderą jaki to, w tak niewielkiej pamięci, można stworzyć efekt. W tych produkcjach najlepsi umieszczali kolorowe plazmy lub też scrolle po sinusie.

Ramos

Moje boje z Keyrah 8-bit

Zapewne wielu użytkowników emulatora C64 chciałoby mieć przed sobą oryginalną klawiaturę od C64, podpiętą do peceta. Normalnie tego nie można wykonać, jednak z pomocą przychodzą nam układy do samodzielnego montażu oraz interfejs Keyrah pochodzący z firmy Individual Computers. Dzisiaj pragnę przedstawić konstrukcję właśnie tej firmy. W tym artykule mowa będzie o wersji 8-bit, która jak nazwa wskazuje przeznaczona jest do współpracy z klawiaturami pochodzącymi z komputerów ośmiobitowych. W tym konkretnym przypadku, podłączona jest do klawiatury C64 i umieszczona w obudowie C64C.

Jak przystało na wspomnianą firmę, całość otrzymujemy zapakowaną w małe estetycznie wykonane pudełko, które kolorystyką nawiązuje bezpośrednio do firmowej strony. W środku znajdziemy interfejs Keyrah, śrubokręt, przewód USB oraz ulotkę formatu A4 spełniającą funkcję instrukcji, w której pomimo swojej skromnej objętości, zawarte są wszystkie potrzebne informacje. Interfejs występuje w wersji 8 i 16-bit. Wersja 16-bit przeznaczona jest dla klawiatur pochodzących od różnych modeli Amig.

Sam interfejs umieszczony jest na niewielkiej płytce, na której umieszczono gniazda



Zawartość zestawu

przeznaczone do podłączenia klawiatury od C128D, C64, VIC-20, C16, diody zasilania komodoraka, Amigi 600 oraz Amigi 1200. Dodatkowo zalecane jest podłączenie dwóch joysticków do odpowiednich portów, bez których

ciężko sobie wyobrazić zabawę z emulatorami lub nawet z oryginalnym sprzętem. Znajduje się także znany z C64 włącznik zasilania, który pełni funkcję przełącznika mapowania klawiatury. Po przełączeniu go w pozycję ON, klawiatura zachowuje się jak normalna z C64 i funkcjonuje prawidłowo pod emulatorem Vice. Po przełączeniu w pozycję OFF, klawiatura zaczyna funkcjonować jak zwykła pecetowa i dzięki niej można pracować pod systemem operacyjnym, który obsługuje standard HID (Human Interface Device). Całe obłożenie klawiatury, łącznie z klawiszami typu Alt, Ctrl, Tab itp. wyjaśnione jest w tabelce w instrukcji. Przyznam się, że po kilku godzinach obcowania z taką klawiaturą, mogłem zapomnieć o pecetowej leżącej pod blatem na wysuwanej półce. Wszystkie funkcje emulatora, które były mi potrzebne (zmiana dyskiety, resetowanie "komodoraka", freezowanie w Action Replay'u itp. rzeczy) wykonywałem za pomocą klawiatury C64.

Obsługiwane joysticki mogą być tylko jednoprzyciskowe oraz bez autofire. Opisane jest to w instrukcji, jednak przytoczę tutaj kilka faktów. Gniazdo nie podaje napięcia +5V, które jest wymagane do obsługi tego układu. Pozostaje więc ewentualna konfiguracja autofire w emulatorze. Dodatkowo nie można podpiąć



Widok na odległość...



Montaż w obudowie C64C

innych urządzeń sterowych znanych z oryginalnego C64, tj. myszki 1351, paddle, light pen, light gun. Wynika to z faktu, że podłączalne joysticki są mapowane jako klawisze bloku numerycznego, co ułatwia ustawienie ich jako joysticków w emulatorze, a nie jako dodatkowe urządzenia widziane pod systemem i osobno konfigurowane w emulatorze. Z opisów znalezionych na internecie można dowiedzieć się, że podniesiono taktowanie całego układu w stosunku do joysticka Competition Pro USB. Spowodowało to lepszą reakcję na wychylenia joysticka i możliwość bezproblemowego grania. Niestety natrafiłem na kilka sytuacji, w których joystick zareagował z pewnym opóźnieniem. Nie wiem czy to wina interfejsu czy może raczej odpowiedzialny za to jest wiek samego joysticka.

Sam montaż płytki jest bardzo prosty i nie wymaga specjalnych wyjaśnień. Producent nie ustrzegł się jednak dwóch drobnych wpadek, które nieznacznie utrudniają ten proces. Pierwszą z nich jest dołączony śrubokręt. Dobrze że jest, jednak jest za mały do śrubek oryginalnych znajdujących się w obudowie C64C. Być może mniejsze są w "mydelnicze". Drugą wpadką jest nie dołączenie jakiegoś przedłuża-

cza do diody POWER w przypadku montażu tego interfejsu w obudowie C64C. Brakuje ok.



Całość w "akcji"

15-20cm dwużyłowego przewodu. Z tego powodu, bez braku możliwości dorobienia takiego kabelka, nie mogę skorzystać z jakże miłego dodatku, którym jest świecenie tejże diody w momencie przestawienia przełącznika w pozycję ON. Praktycznie brakuje w tym przypadku świecącej diody POWER w obudowie C64C, w momencie zabawy z emulatorem.

Nie można jednak mieć jakichś specjalnych zastrzeżeń do samej konstrukcji oraz idei urządzenia. Jest to bardzo ciekawa konstrukcja, za którą należy się uznanie. Polecam ją wszystkim, którym brakuje klawiatury swojego ośmiobitowca lub tym którzy nie mogą odnaleźć niektórych klawiszy z komcia na klawiaturze PC.

Atreus

Zalety:

- możliwość podłączenia klawiatury C64, C128D, C16, VIC-20 do PC
- możliwość podłączenia dwóch joysticków cyfrowych
- prosty i bezproblemowy montaż w obudowie C64 (dopasowane porty)
- bezproblemowe wykrycie w systemie operacyjnym obsługującym standard HID
- bezproblemowa praca pod emulatorem Vice

Wady:

- za mały śrubokręt w stosunku do oryginalnych śrubek
- brak przedłużacza diody LED w przypadku montażu w obudowie C64C
- brak możliwości wykorzystania innych urządzeń sterowych niż joystick cyfrowy

Ratujemy Commodore 64

Patrząc na sprzęt firmy Commodore dochodzę do wniosku że jest raczej niezawodny. Nie oznacza to wcale że się nie psuje. W większości przypadków z winy użytkownika, jego niewiedzy lub zbiegu okoliczności. Do częstych awarii następuje, gdy zasilacz daje za wysokie napięcie. Pomijając sprawę zasilacza, który powinien być szczególnie istotny – a jest traktowany po macoszemu – czasami po prostu coś się i tak sfajczy mimo poprawnych napięć.

Dalszy opis nie należy traktować jako opis serwisowania, a raczej jako uwagi co może czekać potencjalnego serwisanta. Przy odrobinie szczęścia naprawa jest prosta, jednak w

większości przypadków „martwej” płyty naprawiać się nie opłaca, albo trzeba poświęcić dużo wolnego czasu.

Postanowiłem bliżej przyjrzeć się c64. Znalezione schematy w Internecie zawierają niestety drobne, ale istotne błędy. Nie było to jednak wielkim problemem, ponieważ postanowiłem „na żywo” sprawdzić co się stanie gdy nie będzie jakiegoś scalaka na płycie C64. Zanim mogłem to sprawdzić musiałem wszystko wylutować, wlutować podstawki i po kolei wyciągać kolejne podzespoły. Sam proces demontażu wymaga sporo uwagi i cierpliwości gdyż wylutowanie scalaka z dwustronnego druku do najprostszych czynności nie należy. Osoby

które nie mają doświadczenia z demontażem lepiej niech potrenują na czymś innym bo będzie więcej szkody niż pożytku.

Patrząc na ostatni model płyty o oznaczeniu 250469 REV B widzimy 16 układów scalonych czy bardziej specjalizowanych, spełniających różne funkcje. Szczegółowy opis każdego z nich mija się z celem.

Zacząłem po kolei wyciągać kolejne scalaki. Na początek U9 8580 (SID). Odpowiada on za generowanie dźwięku oraz odczyt stanów z potencjometrów podłączonych do wejść CN8 i CN9. Wyjęcie nic nie zmienia poza brakiem dźwięku i obsługi potencjometrów. Jak spojrzymy na schemat, SID ma tylko dwa wejścia do potencjometrów, a mamy możliwość podłączenia ich aż 4 – dwa porty joyów. Za wybór jednego z portów odpowiedzialny jest U18 – 4066. Jego nieobecność uniemożliwia jedynie korzystanie z potencjometrów. Dwa już mam za sobą. Na pewno brak (przepalenie) bezpiecznika nie unieruchomi c64. Wystąpi brak dźwięku, pracy magnetofonu oraz napięć na złączu USER PORT. Bezpiecznik wyjęty. U21 – 4066 – odpowiedzialny jest za przełączanie szyny danych od kolorów. Jego nieobecność jest dość ciekawa, bo obraz dalej jest wyświetlany tylko wygląda jak czarno-biały. Włożyłem go ponownie. Wyjąłem U3 – 74LS08. Zdziwiłem się bardzo, gdy komodorek w ogóle się nie włączył! Na schemacie bramki sumują sygnały i dostarczają na wejścia procesora – AEC i READY. Dodatkowa bramka podłączona do U4 (BASIC & KERNAL) nie jest podłączona, za to ostatnia jest wykorzystana jako bufor sygnału zegarowego – TOD CLOCK – na schemacie jej brak! Teraz wszystko stało się jasne – brak sygnału zegarowego ewidentnie uniemożliwia pracę komputerowi. Włożyłem go z powrotem i wyjąłem U2 – CIA 6526. Odpowiada za magistralę IEC, wybór banków dla VICa oraz USERPORT. Nieobecność powoduje stany nieustalone na VA14 i VA15 (losowe wzory na ekranie). Można wymusić ręcznie te stany i obraz będzie wyświetlany poprawnie, ale to raczej tylko do testów niż ciągłego używania. Włożyłem U2 i wyjąłem U5 – CHAR ROM. Są w nim zapisane tylko znaki graficzne. Nieobecność powoduje bardzo ciekawe efekty na ekranie z powodu stanów nieustalonych (odczytują się „śmieci” zamiast znaków graficznych). Włożyłem go. Pozostał jeszcze U1 – CIA 6526. Odpowiada



za obsługę klawiatury i joysticków. Nieobecność powoduje że C64 nie wyświetla kursora. Jest to spowodowane przez procedurę kernala, która „czeka” na inne dane, a te się nie pojawiają. Z ważniejszych części wszystkie są sprawdzane. Wyjęcie procesora U6 - 8500 unieruchomi komputer – tylko czarny obraz. Brak U7 – VIC 8565 jeszcze bardziej wpłynie na komputer – nie pokaże się obraz oraz nic nie będzie działać z braku sygnału zegarowego. Brak RAMu (U10,U11) także uniemożliwia pracę.

Pozostałe scalaki mają mniejsze znaczenie, co nie oznacza że są niepotrzebne. Najważniejszy jest U20 – 8701 – odpowiedzialny za sygnał zegarowy. U23 – bufor RESTORE, TOD oraz RESET – nieobecność tylko lekko utrudnia pracę – trzeba ręcznie wykonać RESET. U22 - 7406 – bufor magistrali IEC oraz dwóch RESETów oraz IRQ dla U2.

Po pobieżnym opisie widać, że aby komputer prawidłowo działał wszystko musi być sprawne (poza drobnymi wyjątkami). Wiedząc już co jest na pewno potrzebne, aby komputer zaczął jakkolwiek pracę postanowiłem teorię wprowadzić w praktykę.

PIERWSZA PŁYTA 250469 REV B

Po włączeniu jest „czarny obraz”. Sugeruje to sprawność VICa oraz sygnału zegarowego. Dla pewności przełożyłem go do sprawnej płyty – podejrzenia okazały się trafne. Wszystko wskazywało że ROM (BASIC & KERNAL) jest uszkodzony. Po wylutowaniu i przełożeniu do sprawnej płyty okazał się sprawny. Teraz pewnie się zastanawiasz czemu od tego zacząłem. Brak obrazu sugeruje że z ROMu odczytują się błędne dane. Tak mogło być, ale nie w tym przypadku. Jeśli ROM jest sprawny to pozostaje RAM. To był strzał w dziesiątkę. Obie kości były uszkodzone. Pomyślałem, że to już koniec naprawy. Wlutowałem podstawki pod ROM i RAM, włożyłem sprawne kości i... czarny obraz. Coś było dalej uszkodzone. Z wcześniejszych testów wynika, że ani CIA ani CHAR ROM (ich nieobecność) nie unieruchamia komputera. Właśnie ich nieobecność. Zacząłem od U2. Okazał się uszkodzony. W sprawnej płycie komputer nie chciał się włączyć. Jaki jest wniosek? Niesprawność kości objawia się przynajmniej na trzy sposoby: całkowity brak reakcji, błędna reakcja lub... to właśnie jest kluczem do rozwiązania – blokowanie szyny adresowej lub danych. Co akurat CIA blokował ciężko stwierdzić, ale na

pewno blokował. Pomyślałem że pewnie drugi też jest uszkodzony. Po wylutowaniu U1 podejrzenia się potwierdziły. Wlutowałem podstawki i podstawiłem sprawne CIA – dalej czarny obraz. Musiał paść procesor. Wylutowałem i sprawdziłem – działa. Pozostał nieszczęsny CHAR ROM. Okazało się, że jest uszkodzony i blokuje magistralę.

Reasumując. Uszkodzone są: U1, U2 – CIA, U5 CHAR ROM oraz U10 i U11 – RAM. O ile RAM i ROM można czymś zastąpić to CIA już nie. Naprawa jest niemożliwa – chyba że trafi się dawca ;)

DRUGA PŁYTA 250469 REV B

Objaw jak wyżej – czarny obraz. Brakuje SIDA. Bogatszy w doświadczenia postanowiłem zacząć sprawdzanie od innych części. Na pierwszy ogień poszedł RAM – uszkodzony. Podmiana nie daje zmian. Czas na U2 – sprawny. Chyba CHAR ROM padł. Wylutowałem – sprawny. Czas na BASIC & KERNAL. Okazuje się uszkodzony. Procesor zostawiłem w spokoju. Po wlutowaniu podstawek i wymianie uszkodzonych kości RAM oraz ROM z BASIC & KERNAL płyta się włączyła ale ekran pozostał bez kursora. U1 CIA okazał się uszkodzony ale nie blokował szyny.

Jak widać naprawa jest niemożliwa – brakuje CIA.

TRZECIA PŁYTA 250469 REV B

Objaw identyczny – czarny obraz. Zanim zacząłem naprawę byłem mocno zrezygnowany – znowu mnóstwo demontażu. Obejrzałem płytę i jedna z kości była lekko pęknięta. Wyglądało że płyta spadła lub coś na nią spadło. Tu wystarczy wymienić kość.

Ta naprawa była prosta bo było widać fizyczne uszkodzenie.

CZWARTA PŁYTA 250407

Jest to starsza płyta z 1983. Rzut oka i widać że jest na niej „bogato” części. Objaw identyczny – czarny obraz. Z uwagi że płyta była kompletna i nie grzebana, zacząłem od pomiarów napięć – w normie +5V CAN oraz +12V. Jeśli bezpiecznik był by przepalony, to w ogóle C64 nie ruszy. Zanim zacząłem cokolwiek demontować wyjąłem U18 – SID. C64 ożył, ale pokazał 469 BYTES FREE. Tu ewidentnie był problem z RAMem lub sterowaniem. Sprawdziłem RP1 oraz RP2 – wszystkie miały po 33R – są sprawne. Wylutowałem U13 i U25 – 74LS257 – bez zmian. Odpowiadają one za adresowanie kości RAM. Dalsze sprawdzanie po omacku zaprzestałem i napisałem program, który sprawdzał który bit z szyny danych niedomaga – było w końcu prawie pół kB RAMu. Tu wymagane jest małe wyjaśnienie. Ostatnie wersje płyt miały RAM inaczej zorganizowany, czyli 2 kości 4bit x 65536. W starszych płytach kości było 8 – 1bit x 65536.



Pamięć jest dynamiczna i dlatego C64 się włączył ponieważ jedna z kolumn lub rzędów była uszkodzona. Z testowego programu wynikało że bit D2 ma problem. Za to odpowiada kość 4264 (4164) U22. Podmiana i pełna sprawność. Tu widać, że SID sprawia problemy. Diagnoza nie była trudna ze względu na lekkie uszkodzenie.

PIĄTA PŁYTA 250407

Identyczna płyta i identyczny objaw. Napięcia w normie. Po wyjęciu SIDA C64 działa poprawnie.

SZÓSTA PŁYTA 250425

To trochę nowsza płyta. Posiada już procesor 8500. Reszta jest identyczna jak w poprzedniej, przy czym trochę pozmieniało się rozłożenie – numeracja ważniejszych części jest identyczna. Widać było ingerencję poprzedniego właściciela. U5 – KERNAL – był w podstawce oraz wymieniane (lub tylko wylutowane i wlutowane) były U12 – kość RAM – i U25 – 74LS257. Napięcia były w normie. Wyjąłem SID i przełożyłem do sprawnej płyty – blokował uruchomienie komputera. Po jego wyjęciu objaw był identyczny jak poprzednio – czarny obraz. Sprawdzanie zacząłem od wymienianych części. Wylutowałem U12 i U25 – oba scalaki były sprawne. W tym momencie nie wiedziałem co się mogło uszkodzić, więc wylutowałem wszystkie kości RAM. Wszystkie były sprawne. Oba CIA – U1 i U2 – również. Tak samo U3, U4 i U5 – CHAR ROM, KERNAL i BASIC. Procesor – U7 też był sprawny. Pozostał tylko PLA – U17. „Niestety” też był sprawny. Na płycie już praktycznie nic nie zostało. Odłożyłem sprawdzanie na następny dzień. Rzut oka na świeżo utwierdził mnie, że był uszkodzony SID i ktoś go po prostu nie wyjął tylko zaczął demontaż słabym sprzętem bo przelotki od w/w ruszanych części były bardzo wymęczone. Wziąłem multimetr i sprawdzałem przejścia – w dwóch miejscach ich nie było choć wygląd płyty na to nie wskazywał. Teraz na 99% jestem przekonany o marnej jakości demontażu wcześniejszego amatora lutownicy, co zaowocowało przerwami. Wlutowałem podstawki, włożyłem wylutowane części i płyta ruszyła (po usunięciu przerw). Mimo tego mam płytę która prawie w całości jest obsadzona na podstawkami ;)

SIÓDMA PŁYTA 250466

Objaw identyczny jw. Płyta jest z 1986r. Posiada już dwie kości RAM 41464 – reszta jak w pozostałych „starych” płytach. Na pierwszy ogień poszły kości RAM – obie sprawne, potem ROMy – wszystkie sprawne. Z uwagi, że w żadnej płycie nie był uszkodzony procesor pozostały tylko 2 CIA. Niestety jeden z nich był uszkodzony.

Naprawa jest niemożliwa ze względu na brak części zamiennych.

ÓSMA PŁYTA 250469 REV B

Płyta nie była moja tylko dotarła jako uszkodzona. Podobno miał być problem z działaniem joysticków. Aby się o tym przekonać włączyłem testy. Na początek klawiatura, gdyż wykorzystuje te same linie co joysticki. Wszystkie klawisze działały prawidłowo. Przełączyłem test na joystick port 1 – wszystko ok. Na port 2 kierunki działają poprawnie, natomiast po naciśnięciu FIRE komputer się zawiesza. Wg. opisu usterki miało być inaczej. Mimo to postanowiłem coś z tym zrobić. Sprawdzenie przejść oraz filtrów EM wykazała ich sprawność. Uszkodzony był układ CIA. Sprawa była beznadziejna, gdyż znowu naprawa polegała by na wymianie na sprawną sztukę. Nie miałem sprawnej więc rzuciłem okiem na schemat. Za FIRE w port 2 odpowiada pin A4 w CIA. Najprawdopodobniej coś w nim dostało zwarcia itp., że jest problem z wejściowym trybem. Drugi CIA akurat ma pin A4 tylko w trybie wyjściowym jako CLK OUT. Pomyślałem jak by tak zamienić miejscami CIA. Najpierw wylutowałem CIA U1 i włożyłem do innej płyty aby wykluczyć uszkodzenie płyty – było identycznie i komputer też się zawieszał po naciśnięciu FIRE. Wylutowałem drugi CIA U2 i zamieniłem je miejscami. Stacja dysków, klawiatura jak i joysticki działają poprawnie.

Jak widać czasem częściowo uszkodzone części da się jeszcze wykorzystać.

INNE USTERKI

Miałem dwa przypadki w których SID nie grał mimo pełnej sprawności (sprawdziłem w innej płycie). Chodzi o SID 8580. W takim wypadku naprawa nie jest skomplikowana tylko wymaga sprawdzenia kilku prostych rzeczy. Na początek trzeba obejrzeć wszystkie części bierne znajdujące się w pobliżu U9. Okazało się że C77 10uF ma jedną z nóżek wyrwaną z obudowy. Normalnie nie było tego widać. Dopiero przy lekkim wyginaniu usterka wyszła na jaw. Z uwagi że nie miałem takiego kondensatora dałem inny tj. 47uF/16V. Pojemność nie jest krytyczną wartością gdyż to kondensator sprzęgający i jego pojemność wpływa na filtr górnoprzepustowy – po naszymu, jak niskie przepuści częstotliwości.

Drugi taki przypadek nie poszedł tak łatwo – kondensator był w porządku. Bezpiecznik był sprawny mimo to SID nie grał. W takim wypadku należy sprawdzić czy dochodzą napięcia +5V – pin 25 – oraz 9V – pin 28. Napięcie +5V było w normie natomiast +9V w ogóle nie było. To dość ciekawe – SID był sprawny – i nie było mowy, że robi zwarcie. Napięcie wynosiło blisko 0V jednak bezpiecznik się nie przepalił! Pomiar AC przy wyjętym bezpieczniku wykazał że jest około 10,5V~. Wstawienie bezpiecznika powodowało całkowite zwarcie. Rzut oka na schemat i widać że tranzystor Q5 jest odpowiedzialny za stabilizację napięcia +9V. Po wylutowaniu i przemierzeniu tranzystora wydawał się spraw-

ny. Postanowiłem go ponownie nie montować tylko włączyć C64 i sprawdzić jakie jest teraz napięcie AC – dalej blisko 0V. Ten tranzystor był sprawny, ale go w dalszym ciągu nie montowałem tylko zacząłem sprawdzać bliżej. Diody CR3 i CR4 również były sprawne. Napięciem niestabilizowanym +9V zasilany jest też tranzystor Q1 pracujący jako wtórnik dający stabilizowane napięcie dla magneto fonu (silniczka). Wylutowałem go i sprawdziłem – wydawał się być sprawny. Włączyłem C64 i napięcie AC oraz DC +9V niestabilizowane też powróciło. Wlutowałem Q5 i włożyłem SIDA – dźwięk był. Uszkodzony był tranzystor Q1 2SD313. Po wymianie na inny problem ustął i magneto fon też poprawnie działał.

Nie wspominałem o tym, że jak ktoś nie chce od razu rozkręcać C64 gdy nie gra dźwięk warto sprawdzić czy magneto fon „kręci”. Jeśli tak to najprawdopodobniej napięcie +9V również dochodzi do SIDA. Jeśli jednak nie „kręci” to albo uszkodzony jest zasilacz albo przepalony bezpiecznik lub występuje inna usterka.

ZAKOŃCZENIE

Jak widać z naprawą trzeba mieć trochę szczęścia. Nie rzadko uszkodzony jest SID, który całkowicie blokuje pracę komputera. Dość często uszkodzeniu ulegają kości CIA czy RAM – zwłaszcza z ostatnich wersji płyt, które były produkowane głównie w Chinach – niższa jakość i większa awaryjność. Do tego dochodzą kości ROM. W żadnej płycie nie był uszkodzony procesor, VIC czy PLA.

Osiem płyt to za mało, aby z całą stanowczością stwierdzić, że VIC czy procesor zawsze jest sprawny, a jedynie pozwala przypuszczać, że nie uległ uszkodzeniu. Również żaden TTL nie wymagał wymiany. Na pewno na początek należy wyjąć SID, niezależnie od wersji płyty.

Osoby które chcą zdiagnozować swoje płyty powinny posiadać sprawną płytę z wlutowanymi podstawkami.

Wróć jeszcze do sprawy zamienników. W przypadku układów TTL z serii 74xx nie trzeba patrzeć, że oryginał to np. 74LS14 a posiadamy 74HC14. Różnice między nimi są – głównie w szybkości działania – ale przy ślimaczej ich pracy do granicznych parametrów jest bardzo daleko. W przypadku tranzystorów sprawa jest jeszcze prostsza. Zwykle wystarczy patrzeć na to że tranzystor jest typu NPN np. 2SC1815. Zmieników jest bardzo dużo, ale jeśli wiadomo że praca jest jako wtórnik to praktycznie pasuje każdy tranzystor małej częstotliwości małej mocy, jak typowy BC547. Co do tranzystora Q1 2SD313. Tu jest podobnie jak w poprzednim przypadku, tylko tranzystor musi być małej częstotliwości dużej mocy. Świetnie nadaje się do tego BD243.

WYWIAD Z TOMASZEM NIDECKIM

ZNANYM JAKO TONID



Kolejny numer to kolejny wywiad. Tym razem prezentowany wywiad powstawał bardzo szybko, za co dziękuję mojemu rozmówcy. Do jego przeprowadzenia zbierałem się już od dłuższego czasu, ale zawsze coś sprawiało, że do niego jakoś nie dochodziło. Dopiero szukając kolejnej osoby do wywiadu przypomniałem sobie o Tomaszu Nideckim znanym niektórym jako tonid. Można powiedzieć, że był pierwszym Polakiem, który zaczął działać na demoscenie C64, kiedy w naszym kraju jeszcze nie tworzyła się żadna taka scena. Był jedną z niewielu osób w kraju, które korzystały z Fidonetu. Tych i innych ciekawych faktów z życia tonida dowiedziecie się z tego wywiadu.

Kiedy w Polsce jeszcze nikt nie słyszał o demoscenie (szczególnie tej na C64) ty już ostro działałeś na niej. Jak to się stało, że w połowie lat 80-tych zainteresowałeś się demosceną na C64?

Swoją pierwszą Commodore (C64c) dostałem od rodziców jeszcze w 1986 roku — mieszkaliśmy wtedy w Montrealu. Podczas gdy znajomi skupiali się na grach, mnie bardzo szybko znudziła taka forma rozrywki i szukałem bardziej kreatywnych zajęć. Po pierwsze: zacząłem tworzyć muzykę. Tu szczególnie przypadł mi do gustu Rock Monitor 4, w którym można było odtwarzać aż cztery ścieżki (w tym jedną z sampli). Po drugie: chciałem nauczyć się programować, ale programowanie w Basicu mi nie wystarczało. Dlatego też rodzice kupili mi podręcznik C64 Reference Guide, z którego dowiedziałem się o kodzie maszynowym i Assemblerze. Prawdziwym przełomem był jednak zakup modemu i dostęp do BBS-ów. Tak właśnie, za pośrednictwem Prosoft BBS, poznałem niejakiego Bouesa (Sylvaina Bourciera) z grupy Prosoft, który stał się moim mentorem. Nauczył mnie podstaw programowania, a także zaczął wykorzystywać moją muzykę w niewielkich intrach i demach.

Może najpierw zanim porozmawiamy o demoscenie opowiedz jak znalazłeś się w Kanadzie?

Nie jest to żadna szokująca historia. Po prostu mój ojciec pracował w Locie. I trafiła mu się placówka w Kanadzie. Z tego też powodu przenieśliśmy się z rodzicami w Montrealu cztery lata (od 1986 do 1990 roku).

Wyjechałeś w latach 80-tych, gdzie w Polsce istniał jeszcze komunizm. W sklepach brakowało podstawowych towarów, a tam na Zachodzie wszystkiego pod dostatkiem. Jakiego Tobie wywarło to wrażenie?

Byłem chyba zbyt młody, aby zwrócić na to uwagę (wyjechałem w 7. klasie szkoły podstawowej). Trudno mi teraz przypomnieć sobie precyzyjnie odczucia, jakie wywarła na mnie taka zmiana otoczenia, ale przystosowałem się bardzo szybko. Jedyne, co mi zde-

cydowanie przeszkadzało, to w pewnym sensie wyalienowanie: miałem jednego przyjaciela i kilku znajomych, ale dystans między ludźmi „na Zachodzie” był jednak zdecydowanie większy. Przyjemnie zaskoczyli mnie jednak ludzie poznani w BBS-ach. Okazali się o wiele bardziej otwarci i ciekawsi (z mojego punktu widzenia) niż większość znajomych ze szkoły. Z niektórymi z nich mam kontakt do dziś!

Jakie wrażenie wywołał u ciebie kupiony przez rodziców komputer?

Był przeze mnie zamówiony! Mój pierwszy kontakt z komputerem (jeszcze w Polsce, chyba w 1984 roku) miał miejsce u kolegi, który był właścicielem ZX Spectrum. Napisałem u niego swój pierwszy program: miał 127 linijek, a w każdej z nich była komenda BEEP: wynikiem był słynny temat z Gliniarza z Beverly Hills — Axel F. Jednak kiedy zobaczyłem u sąsiada z piętra Commodore 64 (pamiętam do dziś, że zachwyciła mnie muzyka z gry Roland's Rat Race), wiedziałem że to właśnie maszyna dla mnie. Dlatego też, kiedy rodzice zaoferowali mi kupno komputera, doskonale wiedziałem, czego chcę: C64c z C1541 i C1950.

Czyli można powiedzieć, że już od ZX Spectrum zaczęła się twoja przygoda z muzyką na komputerach?

Ponieważ kończyłem podstawową szkołę muzyczną, muzyka zawsze była dla mnie ważna. Dlatego też moje zainteresowania związane z komputerami od początku skupiały się wokół muzyki.

Wróćmy do tematu demosceny C64. Jak to się stało, że wstąpiłeś do dwóch grup kanadyjskich?

Liderzy obu grup (Prosoft Team — Bo-uses i Death — Wasp) byli moimi znajomymi z BBS-ów. Spodobało im się, co robię i zaprosili mnie do siebie.

Jak wyglądała wasza współpraca i jak odbywało się tworzenie produkcji scenowych?

W zasadzie nie było to sformalizowane. Kiedy opracowałem jakiś nowy utwór, podsyłałem go jednemu z nich i czekałem na jego opinie. Zawsze coś ciekawego wymyślili. Wiesz, byliśmy wtedy jeszcze nastolatkami, nie mieliśmy sformalizowanego planu podbicia świata <śmiech>. Co ciekawe, nasza współpraca już wtedy, kiedy jeszcze nie było Internetu, była całkowicie zdalna. Bouesa widziałem może ze dwa razy w życiu, Waspa tylko jeden raz.

W swojej pierwszej kolekcji muzycznej wydanej pod szyldem grupy Death pozdrawiasz swoich kolegów i znajomych z Giełdy Bajka. Czy miałeś może przez ten czas przebywania w Kanadzie z nimi kontakt?



W zasadzie z tylko jednym, Bananem, który swoją drogą był głównym „winowajcą” mojej ewolucji w kierunku Amigi po powrocie do Polski (zobaczyłem u niego Red Sector Megademo i znów — wszystko przez tę muzykę).

Mam rozumieć, że przez ten cały czas w Kanadzie nie miałeś kontaktu ze znajomymi w Polsce i nie wiedziałeś co się dzieje na rodzimym podwórku komputerowym?

Średnio raz na rok odwiedzaliśmy Warszawę, ale w zasadzie znajomości się „urwały”. W podstawówce należałem do typowych nerdów/geeków, a więc moje grono znajomych było i tak niewielkie. Tak więc właściwie kompletnie nie wiedziałem, co się dzieje na polskiej scenie komputerowej.

Czy interesowała cię jako taka demoscena w tym okresie?

Szczerze mówiąc, nigdy nie postrzegłem działalności na demoscenie jako celu samego w sobie. Ważne było dla mnie uczenie się, jak robić fajną muzykę, tworzenie, poznanie nowych technik i technologii. Jeśli jakiegokolwiek moje produkcje trafiały na forum publiczne, było miło, ale nie miałem w tym zakresie większych aspiracji. Z punktu widzenia odbiorcy bardziej niż demoscena interesowały mnie produkcje topowych muzyków z tamtych czasów, takich jak Rob Hubbard czy Ben Daglish. Oni zaś tworzyli muzykę do gier (i był to główny powód, dla którego dużo czasu spędziłem chociażby nad The Last Ninja).

Możesz powiedzieć jak w tym okresie wyglądała taka demoscena w Kanadzie?

THE PROSOFT PRODUCTIONS PRESENT



<< THE PERFECT BEAT HX NO.1 >>

MARITAL CODE: DADY BOOZE! (BOOZE)

BOOZE HERE! OH, THIS IS THE NEW
VERSION OF THE PERFECT BEATHX

HEH: BASEGAM, CLAPS AND
DISPLAY ENHANCED ...

PROSOFT

Trudno Definite Impact nazwać zespołem. To było raczej dwóch geeków grających sobie razem od czasu do czasu w domu (Mark na basie, ja na gitarze i wokalu, C64 na całej reszcie). Nie pamiętam już nazwy oprogramowania, które wykorzystywałem do sterowania MIDI, ale dzięki niemu byłem w stanie tworzyć całe podkłady muzyczne na syntezatorach Kawai K-1 i Yamaha DX-27S. Potem komponowane w tym czasie utwory (to nie były covery) zaaranżowałem od nowa na Amidze w znakomitym programie Dr T's KCS Level II, czego wynikiem było w 1993 roku wydanie płyty: można jej posłuchać pod adresem: <http://soundcloud.com/tonid-2/>.

A czy oprócz muzyki tworzyłeś coś jeszcze na C64?

Wstyd przyznać, ale... napisałem jedno demo. O ile można je tak nazwać. To nieco dłuższa historia. Otóż w BBS-ach opartych na C64 w Montrealu „grasował” troll. Taki prawdziwy troll, zapewne prekursor trolli internetowych. Nazywał się Shadowhawk. Był chamski, agresywny i psuł krew również SysOpom (z tego co pamiętam atakował nawet niektóre BBS-y). Mieliliśmy go tak dosyć, że postanowiliśmy... napisać anty-Shadowhawk demo. Zebrałem informacje (tekstowe) od wszystkich poszkodowanych i stworzyłem w zasadzie prostą „przewijankę” z paroma scrollami, która wyświetlała wszystkie informacje o Shadowhawku — między innymi zdobyte informacje osobowe takie jak jego imię, nazwisko i adres zamieszkania (!). Mimo iż teraz trochę się wstydzę, to demo po rozpowszechnieniu w BBS-ach zrobiło swoje. Troll zniknął gdzieś bez śladu, podobno nawet się przeprowadził (!). Tendencje do walki z trollami i wszelakimi przejawami nieodpowiedniego zachowania w sieci również niestety zostały mi do dziś <śmiech> — patrz <http://spam.jogger.pl>.

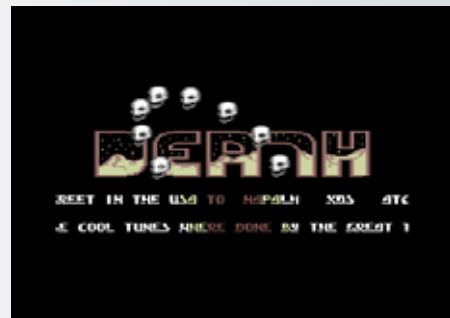


The Perfect Beat Mix

Kiedy przybyłeś do Polski przesiadłeś się na Amigę. Po jakimś czasie zasililiś szeregi grupy Anadune. Jak to się stało, że znalazłeś się akurat w tej grupie.

To kolejna dłuższa historia. Przed wyjazdem z Montrealu zdecydowałem się sprzedać cały zestaw C64, aby w Polsce rodzice mogli mi kupić Amigę (przeprowadzka z komputerem byłaby nieco trudniejsza). Zaczęło się od A500, ale kiedy wyprowadziłem się „na swoje” niedługo potem pojawiła się A1200, następnie A4000 i wreszcie moja ukończona A3000 (z kartą Cybervision 64, Cyberstormem 040, a także ręcznie robioną w Finlandii prototypową kartą muzyczną Delfina DSP!). Chłopaków z Anadune poznałem w... Fidonce (to taka amatorska sieć komputerowa oparta na zautomatyzowanych BBS-ach — pewnie nikt już o niej teraz nie pamięta), gdzie prowadziłem pierwszy w Polsce węzeł sieci na Amidze. Przede wszystkim zaprzyjaźniłem się z Andym, przez którego poznałem Kazika i Acryla. Ponieważ spodobały im się moje kompozycje (nie było ich wiele, ale wystawiałem je, jako niezależny twórca, bodajże na Primavera i Gelloween), zaprosili mnie do grupy. Z tego jednak ko pamiętam pod szyldem grupy nie wydałem chyba żadnego utworu. Wprowa-

dzienie na polską scenę zawdzięcam jednak kochanemu Kubie Husakowi, mojemu najbliższemu przyjacielowi na studiach, a także znakomitemu muzykowi. Podobnie jak ja przed przystąpieniem do Anadune, Kuba zawsze występował się jako niezależny muzyk, w przeciwieństwie jednak do mnie niejednokrotnie wygrywał <śmiech>.



Tonid Musix Collection

Możesz coś więcej opowiedzieć o Fidonecie. Jak dużo osób na nim przesiadywało i jaki sprzęt potrzeba było, aby chodził.

Uczestnicy Fidonetu mogli działać jako punkty lub jako węzły. Różnica między punktami i węzłami była taka, że punkt dzwonił do węzła a nie odwrotnie, czyli węzeł musiał działać tak jak BBS — w określonych godzinach komputer musiał automatycznie odbierać telefon przez modem. Było to jednak znacznie bardziej zautomatyzowane, niż w przypadku BBS-ów. Węzły bowiem automatycznie w określonych godzinach, za pośrednictwem wielopoziomowej struktury, wymieniały się ze sobą całą pocztą. Podczas gdy w przypadku BBS-ów konieczne było podłączenie się do BBS-u i pozostanie online przez cały czas użytkowania (co w Polsce oznaczało ogromne rachunki telefoniczne), Fidonet miał własny protokół wymiany poczty prywatnej oraz czegoś w rodzaju list dyskusyjnych, a każdy użytkownik miał w nim swój cyfrowy adres w formie: <kontyent><kraj-region><węzeł>. <punkt>. Aby więc, kiedy jeszcze byłam punktem (2:480/52.22), np. napisać list do Andy'ego mieszkającego w Opolu, pisałam list w kliencie adresując go do Andy'ego (2:484/2.25) wysyłam go do swojego węzła (2:480/52) dzwoniąc do niego (klient automatycznie wymieniał pocztę po połączeniu), węzeł w nocy wymieniał się pocztą z głównym węzłem warszawskim, ten z kolei tej lub następnej nocy wymieniał się pocztą z głównym węzłem dolnośląskim, tamten z głównym opolskim i wreszcie list po około 2-3 dniach, trafiał do Andy'ego. Nie był to Internet, ale lepszy wróbel w garści! <śmiech>. Do obsługi Fidonetu (zarówno punktów jak i węzłów) istniały specjalne programy, w dużej części darmowe i niezbyt wymagające sprzętowo, istniały nawet programy na C64 (np. QWKie). Fidonet miał jednak jedną wadę: jego struktura wymagała pewnego poziomu organizacji. Istniał np. wybierany zarządca na Polskę oraz podlegli mu zarządcy na pięć rejonów Polski, nie było żadnej ano-

nimowości (obowiązkiem było pisanie pod imieniem i nazwiskiem), istniała „policja” w postaci moderatorów wszystkich konferencji, a to z kolei „skrzywiło” niektórych z nas (jego użytkowników) i prowadziło do różnorakich sporów (w których sam, przynajmniej ze wstydem, zbyt dużo niepotrzebnie uczestniczyłem). W Polsce było nas, użytkowników Fidonetu, jak sądzę około tysiąca.

Rozumiem, że każdy kto posiadał komputer ze złączem RS-232 i odpowiednim oprogramowaniem mógł się podpiąć do Fidonetu? Czy najpierw trzeba było zdobyć odpowiednie pozwolenie, numer, aby móc z tego korzystać?

Nie tyle złączem RS-232 co mode-mem podłączonym do tego złącza <śmiech>. Ale tak, każdy. Wystarczyło tylko znaleźć kogoś, kto prowadzi węzeł i poprosić go o adres punktu (aby skonfigurował w swoim oprogramowaniu węzłowym nowy punkt i przydzielił adres). W moim przypadku był to kolega ze studiów: Jurek Ratajczak zwany Jurgensem. Aby jednak awansować na węzeł trzeba było zdać egzamin (!) z wiedzy technicznej u lokalnego koordynatora regionalnego (którego, tak jak wspominałem wcześniej, okresowo wybierały w demokratycznym głosowaniu pozostałe węzły). Adres węzła był przyznawany przez koordynatora, a następnie wpisywany do oficjalnych list, które były następnie rozsyłane do koordynatorów „wyżej”, albowiem było to konieczne do utrzymania prawidłowo działającej struktury sieci. Ja awansowałem na węzeł (2:480/78) dopiero bodajże po około roku, zakupiłem oprogramowanie do obsługi węzła — Mail Manager (włoskie shareware — zakup nie był łatwy we wczesnych latach 90. — nie było Paypala!) i uruchomiłem pierwszy w Polsce węzeł na Amidze.

Wygląda to na trochę drogą zabawę. Poza tym trzeba było mieć telefon w domu, a w tych czasach był on luksusem i nie łatwo było dostać pozwolenie na jego podłączenie. Może orientujesz ile było osób co używały do tego celu Amigi lub C64 w Polsce.

W latach 90. telefon w domu luksusem? Chyba jednak już nie, to było jednak po upadku komunizmu. Było to o wiele tańsze niż „wiszenie” w necie godzinami na 020 21 22 parę lat później <śmieje się>. Nie znałem nikogo, kto używałby do tego celu C64, były to już jednak „czasy Amigi i Atari ST”. Natomiast amigowców było sporo, na pewno kilkuset spośród tych kilku tysięcy. Większość użytkowników Fidonetu w tych czasach korzystała już jednak z PC.

Porozmawiajmy teraz może o Amidzie. Na twojej stronie zauważyłem kilka kompozycji stworzonych na Amidzie i wystawionych na różnych imprezach komputerowych. Czy jeździłeś na różne takie imprezy?

Tak, ale niezbyt często. Z tego co pamiętam byłem na obu Primaveraach w Starchowicach (94 i 95), Gelloween w Bydgoszczy (94), Intel Outside 2 w Warszawie (95) i czwartej edycji we Włocławku (97), Xenium we Włocławku (99) oraz, po długiej przerwie, na Riverwash w Warszawie (2009), gdzie DJ-owaliśmy wraz z moją ekipą In Dark We Thrust (Lorelein i Tau) — <http://www.idwt.pl>. Swoje utwory wystawiałem bodajże na pierwszej Primaveraze (Cloudhunter), Gelloween (Misthunter), drugiej Primaveraze (Slainte Mhor) i IO4 (Śmierdzi trupem). Próbowałem również swoich sił lokalnie na crazy compo (Flight of the Maya Bee) Oczywiście, nigdy nie wygrywałem <śmiech>.

Czy nadal sprawdzasz na różnych stronach co dzieje się na demoscenie?

Nie, aczkolwiek cały czas mam kontakt (na przykład na Facebooku) z wieloma aktywnymi scenowcami (choćby organizatorami Riverwash), więc od nich dowiaduję się, co się dzieje.

Wspomniałeś, że już w mniejszym stopniu zajmujesz się muzyką. Czyżby ci się już znudziło komponowanie? A może po prostu brak czasu, bo doszły obowiązki dnia codziennego?

Brak czasu. Muzykę tworzyłem na studiach. Teraz jestem już stary <śmieje się>.

mam pracę, rodzinę, obowiązki i wiele różnych innych zainteresowań. Poza tym, wiele lat temu sprzedałem swoje syntezatory i nie orientuję się zbyt dobrze w nowoczesnym oprogramowaniu do tworzenia muzyki. Dlatego też tworzenie muzyki pozostawiam młodemu, sam jestem jej konsumentem.

Na zakończenie tego wywiadu. Co porabia teraz tonid zawodowo i prywatnie?

Zawodowo: od 5 lat jestem tłumaczem-freelancerem, w większości tłumaczę interfejsy i dokumentację dla IBM-a (za pośrednictwem trzech agencji) w programie IBM Translation Manager. Znalazłem swoją niszę, pracuje w domu, w wybranych godzinach, słowem: wolność i niezależność. Prywatnie: mam lat 40, wspaniałą kobietę, wiele różnych zainteresowań (m.in. akwarystyka, RPG-i, muzyka głównie z kręgów psychodelic trance oraz dark independent, ale nie tylko, samochody amerykańskie, okazjonalne czysto amatorskie DJ-owanie), prowadzę na Facebooku stronę „Dobre rady wujka tonida”, sporo gram w WoW-a (śmieje się).

Dzięki za wywiad....

Wywiad przeprowadził....
Ramos



Adapter kasetowy

i problem (małej ilości gier) z głowicy

W początkowej fazie dominacji C64 w naszym kraju, zdecydowana większość jego użytkowników korzystała z dobrze znanego, dedykowanego magnetofonu Datasette. Z oczywistych powodów, na stację dysków mogli pozwolić sobie naprawdę nieliczni (jej cena, jak dobrze wiemy, przewyższała wtedy cenę nowego komputera), a niejeden śnił o niej po nocach. W okresie późniejszym, wraz z postępującym spadkiem owej ceny, liczba użytkowników stacji rosła i nie była już ona aż tak wielkim rarytasem. Jednak mimo to, wysłużony magnetofon wciąż znajdował zastosowanie w wielu polskich domach.

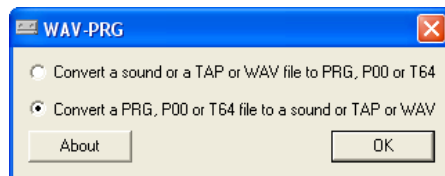
W dzisiejszych czasach natomiast, gdy C64 jest używany z wypiekami na twarzy już tylko przez hobbystów, przyjęło się, że stacja dysków to mus i absolutny standard. Założmy jednak, że to właśnie Ty należysz do wąskiego grona użytkowników Komodorka, które to stacji wciąż nie posiada, bądź też po prostu ze względu na sentyment, w dalszym ciągu preferuje wykorzystanie kaseciaka. Tak, wiem, stacja jest lepsza, szybsza, nie trzeba ustawiać głowicy i takie tam. Ale zanim złapie się za joystick, ta chwila relaksu polegająca na wpatrywaniu się w migające na ekranie czerwono-czarne paski i słuchaniu jazgotu wczytującej się gry, również ma swój urok.

Niezależnie więc od tego, czy jesteś fanem wspomnianych pasków, pisków i zręcznych ruchów śrubokręcikiem, czy też po prostu jesteś na te „przyjemności” skazany z powodu braku innej alternatywy, przedstawię Ci dziś sposób na poszerzenie swojej biblioteki gier przy użyciu pewnego programiku na komputer PC i adaptera kasetowego.

Programem zajmiemy się za chwilę, natomiast zaczniemy od wspomnianego adaptera. Urządzenie to powstało w zamyśle do użycia z radiomagnetofonami samochodowymi, które wciąż są dość popularne. Pozwala ono na podłączenie do takiego radiomagnetofonu praktycznie dowolnego źródła dźwięku. Wygląda to na pierwszy rzut oka, jak zwykła kaseta magnetofonowa, z małą różnicą, ponieważ nie posiada wewnątrz taśmy magnetycznej, a z jej boku wychodzi około metrowej długości kabelek zakończony popularnym wtykiem mini jack. Wystarczy podłączyć go

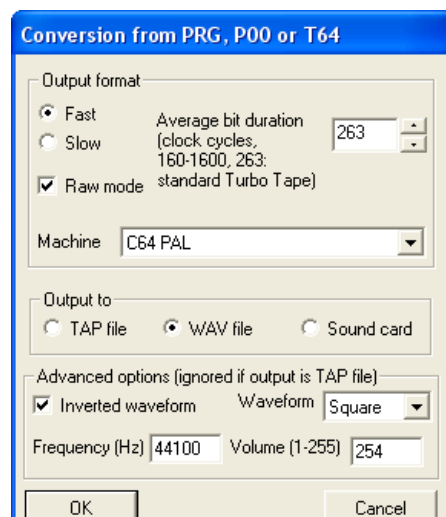
do odtwarzacza CD, MP3, telefonu, czy co tam jeszcze przyjdzie nam na myśl i dźwięk poprzez włożony adapter będzie przesyłany do magnetofonu, który nawet się nie zorientuje, że zaszerwowaliśmy mu taką sprytną podmiannę. Z nabyciem tegoż urządzenia nie powinno być żadnego kłopotu, gdyż jest ono szeroko dostępne, a cena waha się od symbolicznych kilku złotych na aukcjach internetowych, do dwóch, trzech dyszek w stacjonarnych sklepach RTV.

Pora na niezbędne oprogramowanie, a ściślej mówiąc niewielki i banalny w obsłudze programik zwany WAV-PRG (do ściągnięcia chociażby ze strony: <http://wav-prg.sourceforge.net/wavprg.html> – interesuje nas wersja wraz z Audiotap, bowiem zawiera ona dodatkowe pliki DLL, które będą w tym wypadku



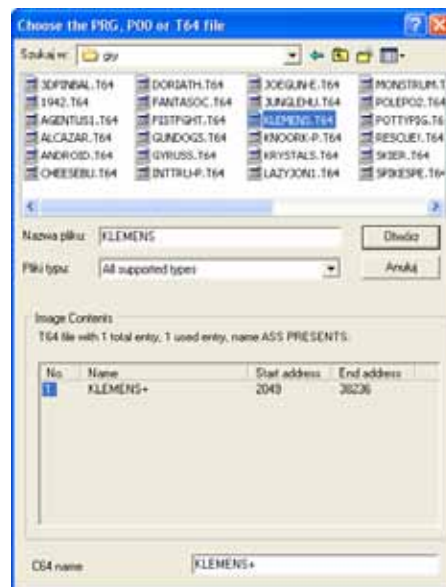
niezbędne). Po instalacji uruchamiamy tenże programik (WAV-PRG) i w okienku startowym wybieramy opcję nr 2, czyli „Convert a PRG, P00 or T64 file to a sound or TAP or WAV”.

W tym momencie pojawia się właściwe okno programu. Zaznaczając „Raw mode” rezygnujemy z umieszczenia fast loadera przed właściwym programem (co oczywiście nie jest potrzebne, jeżeli np. dysponujemy kartridżem zawierającym turbo). Opcja, która nas szczególnie interesuje, to „Inverted waveform”. W zależności od planowanego użycia pliku należy ją zaznaczyć, bądź też odznaczyć – za moment napiszę, od czego to zależy. „Output to” ustawiamy na „WAV file” (można też zapisywać pliku wynikowego, a jedynie jednorazowo wczytać grę na naszym C64). Resztę opcji do standardowego zastosowania proponuję pozostawić bez zmian (oszczędzę sobie w tym miejscu ich opisu, bowiem są dobrze wytłumaczone w dołączonym do programu pliku pomocy). Kronikarskim obowiązkiem wspomnę, że opcja „Machine” po rozwinięciu



daje do zrozumienia, iż program da się analogicznie zastosować do komputerów VIC20 i C16 – nie jestem jednak w stanie przetestować skuteczności tego rozwiązania z powodu braku wymienionego sprzętu.

Po wciśnięciu przycisku OK pozostaje wybrać interesujący nas plik, który poddamy konwersji. Zakładam, iż każdy Commodorec ma ich na dysku swojego PeCeta (wiem, dziwnie to brzmi, ale nie oszukujemy się – dziś jego używanie to niemal konieczność) zgro-



madzonych pod dostatkiem, tak by korzystać z nich przy użyciu emulatora. Jeżeli jednak nie, zdobycie interesujących nas gier w formacie T64 lub PRG zajmie przy użyciu Internetu jedynie chwilę. Wracając do tematu: w górnej części okienka wybieramy interesujący nas plik, w dolnej natomiast, w polu „Image contents” widzimy jego zawartość. Najczęściej jest to tylko jedna pozycja, która zaznacza się automatycznie, gdyby jednak było ich więcej, trzeba to zrobić ręcznie, klikając na numerki przy nazwie (czasami do gry kasetowej może być dołączona instrukcja czy też jakiś obrazeczek jako osobny plik – nie jest on raczej potrzebny; gdy jednak nam na nim zależy – musimy przekonwertować go osobno). Teraz używamy przycisku „Otwórz”. Później już tylko wybieramy nazwę dla wynikowego pliku WAV, klikamy „Zapisz” (screena tego okienka już sobie oszczędzę) i czekamy dosłownie moment – plik został utworzony w podanej lokalizacji.

Mamy już nasze ulubione gry przekonwertowane do formatu WAV, co więc teraz? Opcje ich użycia są właściwie trzy. Dwie pierwsze przy użyciu wspomnianego adaptera: możemy wczytywać gierki bezpośrednio łącząc adapterem Datasette z wyjściem „line out” karty dźwiękowej komputera PC i puszczając je na jakimś odtwarzaczu, bądź też nagrać pliki WAV na jakieś urządzenie przenośne (np. odtwarzacz MP3) i to je połączyć z Datasette. No i właśnie w zależności od tego, czy będziemy zapis audio naszej gry odtwarzać prosto z Pe-Ceta, wspomniana wcześniej opcja „Inverted waveform” powinna być zaznaczona, gdy z kolei planujemy przenieść plik na inne urządzenie i dopiero stamtąd odpalić – „Inverted waveform” powinno pozostać odznaczone, by gra wgrała się prawidłowo. Trzecią możliwość

zostawimy sobie bonusowo na koniec.

Próbując zamontować adapter kasetowy w kieszonce naszego magnetofonu można się odrobinę zdziwić, bowiem niestety wychodzący z boku kabelek zdecydowanie przeszkadza w tym zadaniu. Zamiast wydłubania dziurek czy robienia jakichś innych strasznych rzeczy pocziwiem Datasette, proponuję po prostu wyjąć klapkę. Trzeba wygiąć nieco jej zaczepy, tak by wyszły z trzymających je otworów, jednak zrobić to z wyczuciem, nie narażając plastiku na połamanie.

Oczywiście jak to przy zabawach kasetskami, przed rozpoczęciem wgrywania pozostaje jeszcze kwestia ustawienia głowicy. W tym wypadku jednak, zamiast kręcenia śrubokrętem (nie powoduje to żadnej różnicy) kluczowe jest ustawienie odpowiedniego poziomu głośności na urządzeniu przesyłającym sygnał przez adapter. Od tego właśnie parametru zależy, czy wgrywanie ma szansę zakończyć się sukcesem. Nie muszę chyba mówić (na wszelki wypadek powiem, co mi tam),

że wzorcem jakości są dwa cieniutkie paski bez żadnych dodatkowych zakłóceń. Jeżeli więc program do ustawiania skosu głowicy pokazuje nam „brud”, bądź też nie pokazuje nic, a sygnał na pewno dochodzi – bawimy się głośnością urządzenia przesyłającego ten sygnał. Taka mała uwaga: jeżeli w tym momencie zamiast dobrze znanych dwóch paseczków widzimy trzy, spróbujmy jeszcze raz przygotować interesujący nas plik WAV, tym razem przeciwnie ustawiając opcję „Inverted waveform”, bo prawdopodobnie w tej kwestii leży problem (zdarzają się karty dźwiękowe, które mimo tego co napisałem wcześniej, nie wymagają jednak odwracania sygnału).

Zaraz, zaraz... pomyślą jednak Ci, którzy dobrze pamiętają rytuał wgrywania gierki z kaseciaka. Przecież po zgłoszeniu się nazwy gry silniczek Datasette zatrzymuje się i komputer nasz oczekuje na wciśnięcie spacji, by kontynuować wczytywanie. Co zrobić przy stałym źródle dźwięku? Rozwiązanie jest banalne, ale wymagające odrobinki refleksu: po prostu czekając z palcem na klawiszu spacji, szybko ją



Obraz 1 – trzeba podregulować poziom głośności w urządzeniu przesyłającym dźwięk

Obraz 2 – trzeba przeciwnie ustawić opcję „Inverted waveform” przy konwersji gry na WAV

Obraz 3 – wygląda na to, że wszystko jest w porządku, można rozpocząć wgrywanie.

w tym momencie wcisnąć. Jest to chyba jedyna niedogodność przy korzystaniu z tego rozwiązania, ale da się do tego przyzwyczaić.

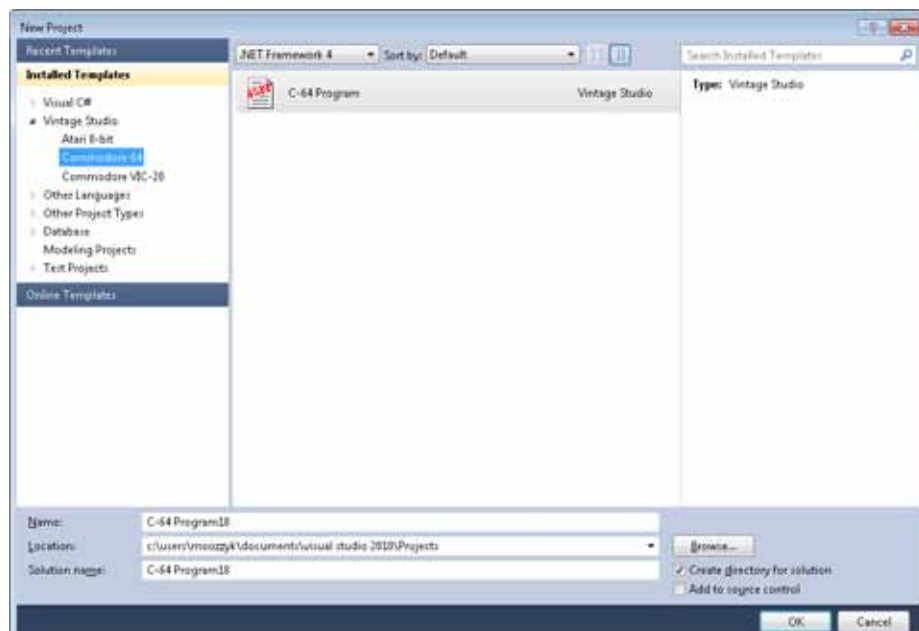
Większość odtwarzaczy MP3 bez problemu odtwarza także pliki WAV, o których cały czas mowa, tak samo chociażby telefon. Czy można jednak używać plików MP3, skompresowanych już we własnym zakresie z wcześniej posiadanych WAV? Otóż z testów moich wynika, że przy dość wysokim bitrate (256-320kbps) wszystko działa bez zarzutów, problemy zaczynają się z plikami mocniej skompresowanymi (swoją drogą, pliki MP3 wymagają nieco niższego ustawienia poziomu głośności niż WAV, trzeba też pamiętać, by pozostały MONO). Biorąc jednak pod uwagę, że duża jednoplirkówka w formacie WAV, przy czasie odtwarzania ok. 2 min (w systemie turbo oczywiście) zajmuje jakieś 5MB, czy jest w ogóle sens kompresji? Jeżeli komuś zależy, może oczywiście próbować, ale chyba nie ma po co. Można też eksperymentować z innymi formatami kompresji dźwięku (dla tych, którym się wybitnie nudzi w życiu).

Jeżeli ktoś ma w domu sprzęt audio wyposażony w magnetofon z możliwością nagrywania na kasetę sygnału z wejścia linowego (czyli mówiąc najprościej – chociażby standardowa „wieża”, ale nie tylko), może po podłączeniu do niego komputera (najczęściej kablem mały jack – 2x chinch) wykonać sobie kasetę pełną ulubionych przebojów i później korzystać z niej najnormalniej w świecie w Datasette bez żadnego adaptera, zupełnie jak za starych dobrych czasów. To właśnie wspomniany wcześniej trzeci sposób wykorzystania programiku WAV-PRG. Przy braku wymaganego do tego celu sprzętu audio można też kombinować w taki sposób, by najpierw wgrać gierkę do pamięci Komodorka przy użyciu adaptera, a następnie zapisać ją na czystej kasiecie umieszczonej w Datasette komendą „SAVE”.

Dzięki opisanym sposobom, przy wykorzystaniu niewielkiej ilości sprzętu otrzymujemy dostęp do nieograniczonej biblioteki gier, które możemy z powodzeniem odpalać na naszym prawdziwym Komciu (emulator to wygodna sprawa, ale odczucia towarzyszące zabawie na realnym C64 są niezastąpione) i bawić się do woli. Nie musimy też obawiać się, że posiadana kaseta z ukochanym tytułem kiedyś się rozmagnesuje i stracimy bezpowrotnie szansę na ponowne jej użycie. I najważniejsze, znów zaznamy stanu hipnozy wywołanego przez dobrze znane migające paski, w oczekiwaniu na wieńczący dzieło i oznaczający ostateczny triumf komunikat „LOADING – READY” (są tacy, którzy to lubią i nie ukrywam, że ja do nich również się zaliczam). Ale pamiętajcie: „nie ruszajcie joystickiem, bo się nie wgra”.

noctropolis

VINTAGE STUDIO



Rys. 1

Vintage Studio, to oparte na Visual Studio 2010 zintegrowane środowisko programistyczne dla komputerów retro, umożliwiające kompilację i uruchamianie programów bezpośrednio z Visual Studio 2010 (i 2012). W chwili obecnej wspierane platformy to Commodore 64, Vic-20 oraz „małe” Atari.

Vintage Studio składa się z dwóch rozszerzeń (vsix) do Visual Studio. Pierwsze rozszerzenie służy do zarządzania projektem oraz jest odpowiedzialne za integrację kompilatora i emulatora z Visual Studio. Drugie rozszerzenie to edytor.

Project system dodaje do Visual Studio nowe szablony (rys1) projektów dla „supported” platform (w tej chwili Commodore 64, Vic-20, „małe” Atari) umożliwiające tworzenie nowych projektów.

1. Tworzenie nowego projektu - platformy wspierane przez Vintage Studio

Projekty wiążą pliki źródłowe z kompilatorem i umożliwiają uruchomienie skompilowanego programu w emulatorze. Dzięki zastosowaniu takiej (jedynej i słusznej) architektury praca z projektami Vintage Studio nie różni się od

pracy z innymi rodzajami projektów dostępnymi w Visual Studio, jak projekty w C# czy VB.NET. Osoby przyzwyczajone do skrótów klawiszowych będą dalej mogły z tych skrótów korzystać i będą one działać w ten sam sposób jak w innych projektach. Błędy kompilacji pojawiają się, zgodnie z oczekiwaniami, w okienku z błędami (rys. 2).

2. Błędy kompilacji wyświetlane są zgodnie z oczekiwaniami w okienku z błędami.

Ctrl+F5 uruchomi program w emulatorze, a F5 (w zależności od możliwości emulatora) uruchomi program w trybie debugowania (rys. 3).

3. F5 uruchamia program w trybie debugowania zatrzymując się na pierwszej instrukcji.

Edytor umożliwia wygodną edycję programu w Visual Studio. Pierwszą, rzucającą się w oczy rzeczą jest kolorowanie składni (rys. 4) – mnemoniki, etykiety, komentarze czy liczby mają różne kolory (jest to szczególnie przydatne w przypadku liczb, gdzie liczby zapisane w różnych systemach liczenia mają różne kolory, co ułatwia wychwycenie subtelnych błędów (np. że napisało się lda #10 zamiast lda #\$10)).



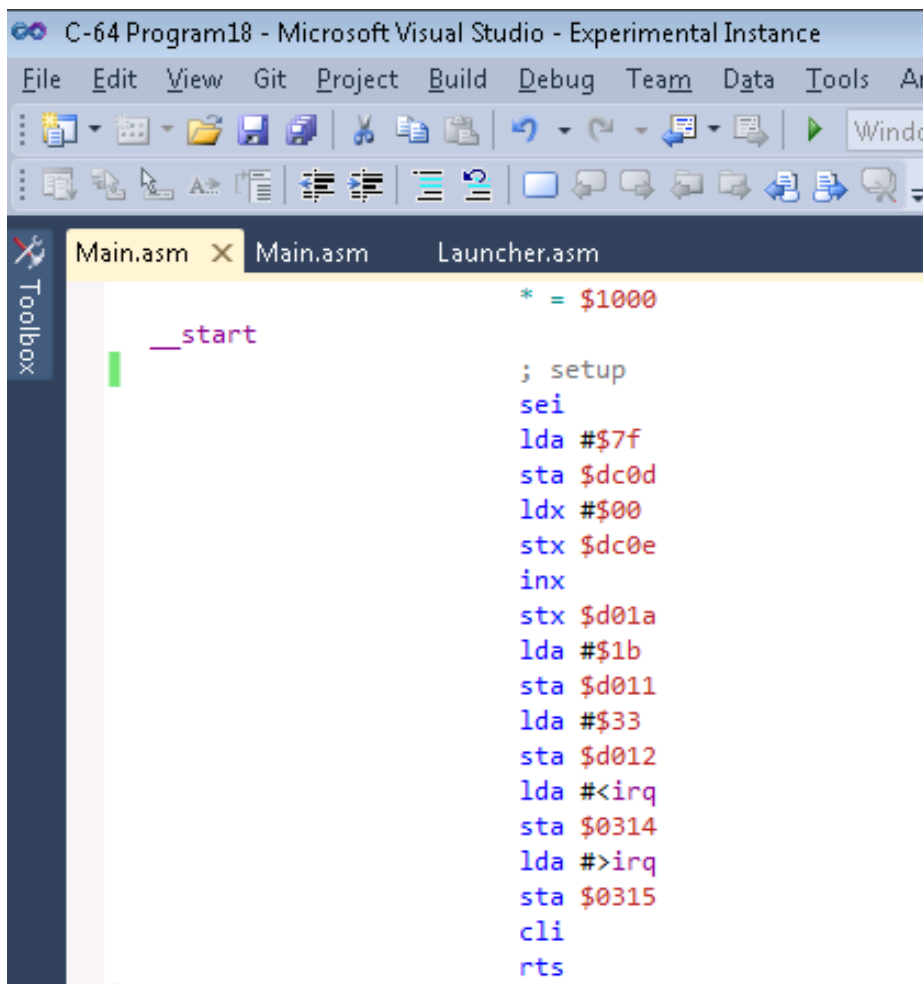
Rys. 2

4. Kolorowanie składni ułatwia kodowanie.

Standardowe kolory można zmienić z menu (Tools > Options, Environment > Fonts and Colors, elementy zaczynające się na „Visual64 Tass” w oknie „Display Items”). Kolejną przydatną funkcjonalnością jest formatowanie kodu. Funkcja ta działa w dwojaki sposób: po pierwsze formatuje istniejący kod (Edit > Advanced > Format Document, Ctrl+E,D). W tym trybie linie z instrukcjami są wyrównywane tak, aby zaczynały się w tej samej kolumnie, etykiety przesuwane są do początków linii, a komentarze są formatowane w zależności od kontekstu.

Drugi tryb to interaktywne formatowanie kodu (tzn. podczas pisania). W tym trybie kod jest formatowany po naciśnięciu klawisza Enter (podobnie jak to robił Turbo Assembler na C-64). Inną funkcją edytora jest komentowanie bloków kodu – wystarczy zaznaczyć fragment kodu i wybrać Edit > Advanced > Comment Selection (Ctrl+E, C) i wszystkie zaznaczone linie (lub bieżąca linia jeśli nic nie zostało zaznaczone) zostaną „zakomentowane”. W podobny sposób można „odkomentować” kod za pomocą opcji Edit > Advanced > Uncomment (Ctrl+E, U).

Vintage Studio wymaga co najmniej edycji „Professional” Visual Studio 2010. Visual Studio nie jest darmowe i nie każdy ma do niego dostęp. Nie powinno to jednak zniechęcać bo... Microsoft niedawno udostępnił wersję RC Visual Studio 2012 (disclaimer: tak, maczałem w tym paluchy). Wersja RC jest dar-



Rys. 4

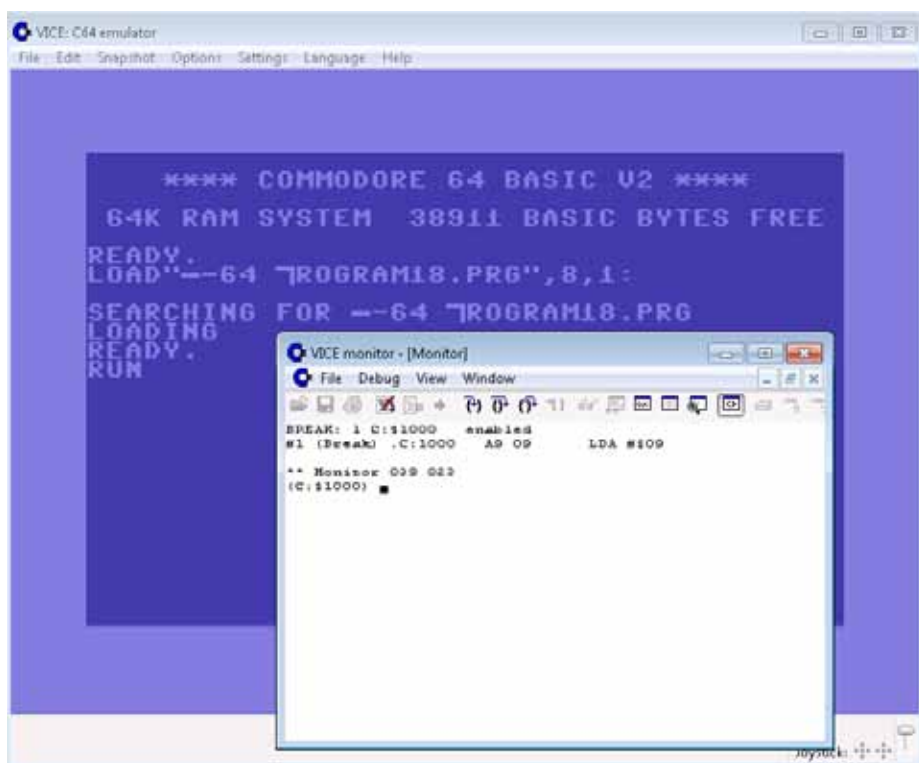
mowa i można ją ściągnąć ze stron Microsoftu. Vintage Studio powinien z nią działać (nie miałem jeszcze okazji dobrze jej przetestować,

ale nie widzę powodu dla którego Vintage Studio nie miałoby działać z Visual Studio 2012).

Dodatkowo Vintage Studio wymaga kompilatora 64Tass. Można go ściągnąć z <http://singularcrew.hu/64tass/>. Na stronie można także znaleźć przydatne instrukcje dotyczące składni etc.

Do uruchamiania programów potrzebny jest emulator. Dla programów przeznaczonych dla C-64 i Vic-20 standardowo skonfigurowany kompilator to Vice (<http://www.viceteam.org/>). Programy dla „małego Atari” odpalane są w emulatorze Atari800Win (http://atariarea.krap.pl/PLus/index_us.htm). Vintage Studio jest projektem open source’owym. Binarki i źródła wraz ze szczegółami dotyczącymi instalacji oraz konfiguracji znajdują się na githubie: <https://github.com/moozyk/VintageStudio>. Na YouTube można znaleźć krótkie video przedstawiające Vintage Studio: <http://www.youtube.com/watch?v=zdE5usXKyIk>. Mój blog także zawiera krótki post na ten temat: <http://blog.3d-logic.com/2012/05/12/introducing-vintage-studio/>.

Paweł „moozyk” Kadłuczka



Rys. 3

Commocoffee 64

Najzwyklejsza czarna, zbożowa, espresso, cappuccino albo latte? Rodzajów kaw i sposobów jej przyrządzania jest naprawdę mnóstwo, bo i wielu jest na świecie wielbicieli tego napoju, uznawanego niekiedy za używkę (właściwie to jest nią bezdyskusyjnie, ale zdarzają się i tacy, którzy nie przyjmują tego do wiadomości). Jedni piją, by delektować się smakiem, inni by na skutek zawartej w kawie kofeiny włączyć w siebie trochę energii i postawić się na nogi, gdy czeka na nich jakieś wyjątkowo ciężkie zadanie.



Ano właśnie, wyobraźmy sobie następującą sytuację: siedzimy przed monitorem naszego komputera, dobiega 22-ga, mamy jeszcze tyle pracy do zrobienia, a nam dosłownie kleją się oczy. W takim i podobnych przypadkach, napój o którym mowa może stać się prawdziwym wybawieniem.

Trzeba wtedy pobiec do kuchni i odrywając się od pracy rozpocząć „rytuał” parzenia. A co, jeżeli po prostu nam się nie chce, albo też boimy się, że w czasie tej kilkuminutowej przerwy wypadnie nam z głowy coś ważnego? Jest rozwiązanie, bo można to zrobić inaczej – nie oddalając się od sprzętu, na którym właśnie pracujemy. Jak? Obecnie przez port USB można podłączyć do komputera prawie wszystko. Złącze jest bardzo popularne, a producenci gadżetów prześcigają się w pomysłach. Nic więc dziwnego, że wśród tych ciekawostek znaleźć można także ekspres do kawy.

My tu sobie gadu gadu o kawce i współczesnych komputerach, ale co z tego właściwie wynika i co to ma wspólnego z C64? Chodzi o fakt, iż w czasach królowania komputerów 8- i 16-bitowych funkcję portu USB pełniło złącze RS-232. W C64 wbudowano tzw. User Port – programowalny port wejścia-wyjścia, mogący służyć za port standardu RS-232. To dzięki niemu można było podłączyć do C64 modem, drukarkę czy ploter (nie była to jednak jedyna opcja dla peryferii, o czym za chwilę). No

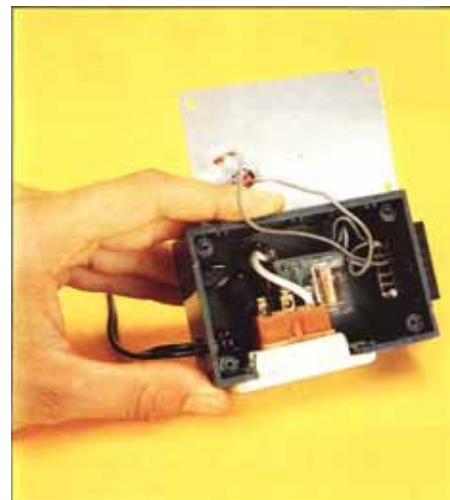
i podobnie jak dzisiaj, oprócz standardowych urządzeń do zastosowań czysto informatycznych znajdziemy także rozwiązania, które nie mieszczą się w tej kategorii. Była mowa o ekspresie do współczesnych PC-tów, ale mało kto może wie, że podobne urządzenie istniało już jakiś czas temu i co więcej, można było podłączyć je do naszego Komodorka. Takim właśnie urządzeniem był stworzony we Włoszech Commocoffee 64, tj. ekspres do kawy (!) podłączany właśnie przez User Port lub Cassette Port do komputerów C64 i VIC 20!

Urządzenie, które jest tak naprawdę zestawem do samodzielnego montażu, można przystosować do pracy zarówno poprzez User Port, jak i port magnetofonu (który, jak widać, poza zapewnieniem współpracy z dedykowanym Datasette mógł również pełnić inne ciekawe funkcje). Istnieją zresztą schematy obrazujące oba te rozwiązania, konstruktorzy polecają jednak tą drugą wersję (to ona znajduje się właśnie na reklamowych zdjęciach), jako że jest to sprawa konstrukcyjnie prostsza (nie wymaga zastosowania tranzystora) i nie obciąża tak C64. O tym, czym dokładnie była i jak działała ta cała konstrukcja, zajmiemy się za chwilę.

Już na samym początku trzeba niestety wytknąć opisywanemu urządzeniu spory minus. O ile terazniejsze, podpinane pod port USB konstrukcje pozwalają na jednoczesną i swobodną pracę na komputerze w czasie działania urządzenia, o tyle niestety Commocoffee 64 wymagało od użytkownika Komcia by zaprzestał na ten czas pracy, bowiem działający ekspres całkowicie przejmował kontrolę nad



sprzętem. W tym czasie można było jedynie czekać na zwolnienie zasobów, ponieważ systemy tworzone dla C64 / VIC20 nie posiadały wielozadaniowości i uniemożliwiały równoczesne wykonywanie więcej niż jednego procesu (dodajmy, że tak działa również MS DOS). Jeśli więc chciało się zaparzyć kawę z Commocoffee 64, trzeba było czekać na zagotowanie się wody. Dopiero po tym można było zabrać się za pracę lub zabawę z wykorzystaniem naszego C64. Rozwiązaniem stawała się więc chwila relaksu i np. liczenie much na ścianie,



lub po prostu... podłączenie sprzętu do innego C64 (jeżeli oczywiście był akurat pod ręką).

Konstrukcyjnie Commocoffee 64 jest po prostu zwyczajnym kompaktowym ekspresem do kawy działającym podobnie jak tradycyjny odpowiednik. Nie da się oczywiście pominąć faktu podłączenia do Komodorka, ale tak naprawdę jedyną funkcją, jaką spełniał w tym wypadku komputer było zastąpienie timera, którego sam ekspres był pozbawiony. Ów timer, to po prostu krótki BASIC-owy program, kontrolujący za pomocą POKE'ów włączanie i wyłączanie napięcia. Napięcie 5V jest podawane na odpowiednie gniazdo, wtedy na uzwojeniu wytwarza się pole elektromagnetyczne, dzięki któremu zwierają się ścieżki doprowadzające 220V do podstawki, która podgrzewa pojemnik z kawą. Zainteresowanych tematem i spragnionych nieco większej ilości szczegółów ze strony czysto technicznej odsyłam do włoskiego czasopisma Mikrokomputer (a konkretnie numeru 42/1985), gdzie ukazał się dość wyczerpujący opis tego właśnie urządzenia (jego zeskanowana wersja krąży zresztą po Internecie) – wymagana jednak będzie naturalnie znajomość języka włoskiego, bądź zabawa z translatorami. Jak było już wspomniane, nasz ekspres oprócz C64 równie udanie współpracuje także z VIC-20, jednak do tego celu konieczna jest oczywiście odrębna wersja oprogramowania (listingi obu programików w przywołanym chwilę wcześniej Mikrokomputerze).

Czy był i jest zatem sens, by używać opisywanego urządzenia – ciekawostki zamiast rozwiązań bardziej standardowych? A owszem, bowiem kawa, w której przygotowaniu maczał palce (a raczej obwody) nasz C64 musi smakować naprawdę wybornie.

noctropolis & Ramos



EasyFlash 3 to następca dobrze znanego w środowisku entuzjastów C64 kartridża EasyFlash. W przeciwieństwie do standardowych kartridżów, umożliwia on zaprogramowanie go bezpośrednio z poziomu Komodorka. W przypadku trzeciej odsłony mamy do czynienia z siedmioma przełączanymi slotami, z których każdy zachowuje się jak w pełni funkcjonalny EasyFlash 1. Nasuwa się jednak pytanie: jak trójka może być następczynią jedynki, czyli gdzie podziała się zatem wersja oznaczona numerkiem 2? Odpowiedź jest prosta: nie wyszła ona po prostu nigdy poza fazę prototypu.

Wracając do tematu: w przypadku EasyFlash 3 mamy tak jakby do użytku siedem urządzeń w jednym. Za pomocą menu wybrać możemy, które z nich odpalamy (a pojemność każdego slotu to 1MB). Dodatkowo posiada on jeszcze port USB, który umożliwia

jego zapis bezpośrednio przy połączeniu z PC. Przy użyciu kartridża można również zastąpić oryginalnego, wewnętrznego kernala C64 innym (obsługuje do ośmiu różnych), zachowując kompatybilność z oprogramowaniem w dokładnie takim stopniu, gdyby był on wbudowany w Commodore. Jakby tego jeszcze było mało, EF3 daje także możliwość emulacji kartridżów z freezerem, jak np. Action Replay. Możliwości wykorzystania tego sprzętu jest więc naprawdę bardzo dużo.

Warto opisać przy okazji historię powstania EasyFlash'a. Jego twórca chciał po prostu na początku we własnym zakresie odtworzyć jeden z oryginalnych, zawierających zestaw gier, kartridżów wydanych przez Ocean i podarować pewnej osobie w prezencie świątecznym. Przeanalizował więc budowę tegoż kartridża, ale jako że nie miał programatora EPROM i lampy UV, a nie chciał się w nie spe-

cialnie zaopatrywać, postanowił użyć niejako w zastępstwie pamięci flash. Wkrótce rozszerzył swoje wymagania i pojawiła się również dodatkowo chęć możliwości zapisu kartridżów 8k, 16k i Ultimax. Za tym poszły kolejne pomysły i stworzyła się idea skonstruowania czegoś poważniejszego. Prace nad EasyFlash'em stały się faktem. Założeniem towarzyszącym budowie kartridża było też użycie jak najłatwiej dostępnych i możliwie najtańszych komponentów (w przypadku wykonania we własnym zakresie, jest to według wyliczeń twórcy koszt około 20-stu euro).

Projekt EasyFlash to oczywiście nie tylko samo urządzenie, ale oczywiście niezbędne oprogramowanie pozwalające wykorzystać we właściwy sposób jego możliwości. Opisując w skrócie: EasyProg to program odpalany bezpośrednio na C64 i umożliwiający zapis pliku .crt (znajdującego się np. na dysku w stacji 1541 lub na karcie SD w SD2IEC) na kartridżu. W zależności od wielkości pliku zapis taki może potrwać od minutki do około dwudziestu. Za jego pomocą umieścimy również na kartridżu nasze zamienne kernale. Przeznaczona z kolei dla komputerów PC aplikacja Draco Cart Marker pozwoli właśnie stworzyć wspomniane pliki .crt, będące kompilacjami naszych ulubionych gier czy też programów w formacie .prg lub niektórych formatów już istniejących obrazów kartridżów. Użycie EasySplit będzie z kolei niezbędne w sytuacji, gdy nie posiadamy napędu, który jest w stanie jednorazowo pomieścić dużego pliku .crt – jak sama nazwa wskazuje, program pozwoli go skompresować oraz podzielić na części. Od niedawna jest również dostępne oprogramowanie pozwalające na połączenie EF3 z komputerem PC za pomocą USB i dokonywanie wszelkich niezbędnych operacji z poziomu tego drugiego sprzętu.

Na stronie internetowej <http://www.skoe.de/easyflash/>, oprócz mnóstwa informacji odnośnie całego projektu znajduje się dokładna instrukcja jak zmontować urządzenie, a wszystko jest wyczerpująco opisane i okraszono-



Ekran startowy Easy Flash 3



Zmontowana płytką z Easy Flash 3

ne pomocnymi zdjęciami. Znajdują się tam też oczywiście dokładne schematy budowy, jeżeli ktoś czuje się na siłach i zamiast zamówienia zestawu montażowego wybierze zbudowanie wszystkiego od podstaw – to już jednak „wyższa szkoła jazdy”.

Głównym pomysłodawcą oraz twórcą całego przedsięwzięcia jest Thomas „skoe”



Zestaw do samodzielnego montażu



Programowanie na Commodore 64

Giesel. Jak sam jednak mówi, bez pomocy wielu osób (które to wymienia na swojej stronie) stworzone urządzenie i współpracujące z nim oprogramowanie nie prezentowało by tak wysokiego poziomu. Z tego też powodu jest im bardzo wdzięczny za pomoc. Ważnym odnotowania faktem jest także to, iż cały projekt

został opatrzony licencją open source, by wszyscy zainteresowani mogli zapoznać się z najgłębszymi szczegółami technicznymi budowy i kodu oraz ewentualnie następnie z tej wiedzy skorzystać. Jeszcze tylko słówko na temat możliwości zakupu EasyFlash'a 3: gotowy do montażu zestaw można nabyć za kwotę 64 euro w tym oto miejscu: <http://retro-donald.de> (oczywiście cena ta jest aktualna w chwili pisania tego tekstu i może z czasem ulegać zmianie).

noctropolis

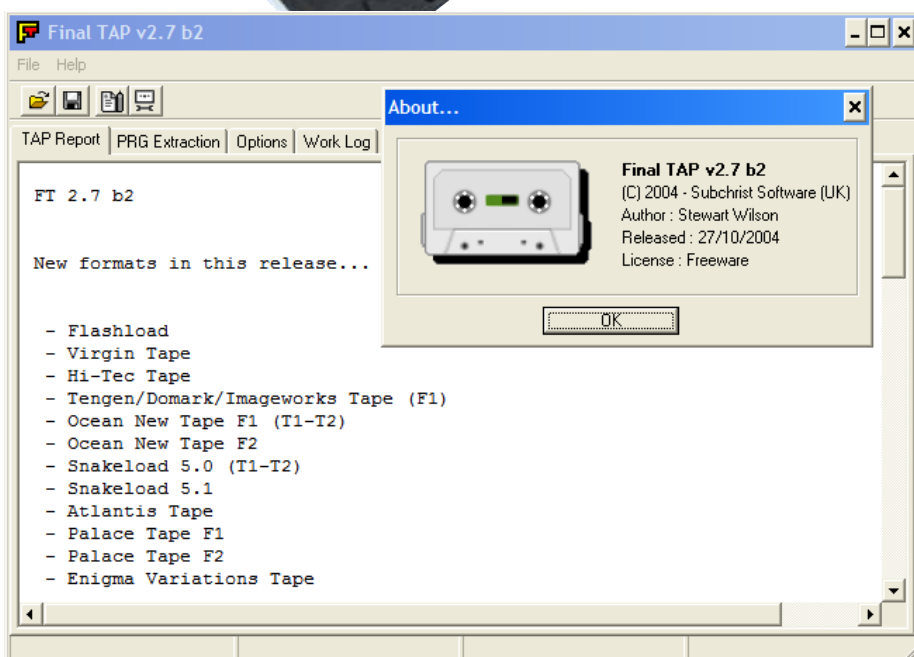
Zdjęcia w artykule pochodzą ze stron:
<http://skoe.de/easyflash/>
<http://ilesj.wordpress.com>
<http://blog.c128.net>

LOADERY TAŚMOWE cz. 1

Dziś chciałbym przedstawić Wam sposoby na wyciąganie gier lub programów zapisanych na kasetach. Do wykonania tego pomocny okaże się program Final Tap lub emulator. Na Commodore 64 stosowana około 50 różnych sposobów zapisu danych na taśmie. Część z nich korzystała z najpopularniejszego formatu zapisu jakim był TPRG (to ten format dzięki któremu czytywało się gry za pomocą np. Black Box-a lub ABC TURBO). Pozostałe formaty zapisu stosowały własne loadery. Z początku twórcy gier stosowali standardowy komodorowski sposób ładowania co w późniejszym czasie spowodowało powstanie wielu znacznie szybszych sposobów ładowania danych z magnetofonu. Niestety niektórzy dodawali obrazek lub muzykę, co w wielu przypadkach spowalniało proces czytowania, znacznie go wydłużając. Każda oryginalna gra czy program zapisana na kasecie czytuje się poleceniem LOAD. Przedstawione poniżej przykłady mają jedynie pokazać jak bez zmieniania kodu pograć w oryginał np. z dyskiety zamiast kasety.

Zanim jednak przystąpimy do działań opiszę program z którego skorzystałem do uzyskania osiągniętego celu. Do „wyciągnięcia” oryginalnych gier zapisanych tym sposobem, posłużyłem się programem Final Tap. Po otwarciu pliku rozpocznie się proces rozpoznawania nagrania, który może trwać dłuższą chwilę. Program zawiera on 4 karty:

- **Tap Report** - wyświetla informacje na temat daty utworzenia nagrania, jego rozmiaru, czasu trwania oraz rozmiarów poszczególnych segmentów,
- **PRG Extraction** - pokazuje wszystkie rozpoznane nagrania na kasecie. W kolumnie Source wyszczególnione są rodzaje nagrania, Start – początek danych, End – koniec danych, Size – rozmiar i suma CRC32. Dodatkowo można dodać ofset (przesunięcie) na podglądzie szesnastkowym z prawej strony lub włączyć dis aseblację. Na samym dole znajduje się opcja Unify sequential neighbours czyli łączenia ze sobą następujących po sobie segmentów. Istnieje też możliwość zmiany rozmiaru czcionki podglądu z prawej strony. Aby to zrobić naciśnij i przytrzymaj lewy przycisk myszki i użyj rolki,



- **Options** – tu możesz wybrać rodzaje loaderów podczas sprawdzania nagrania, tolerancję odczytu, ustawienia optymalizacji czy ignorowania błędów,
- **Work Log** – czyli status wykonanych operacji.

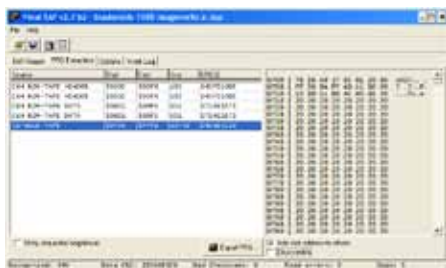
Nas będzie interesować PRG Extraction. Podczas eksportowania plików do formatu PRG program dodaje na końcu numer porządkowy wybranego pliku. Przykładowo dla pliku quedex-1987-thalamus-a-.tap i wyeksportowaniu 5 pliku przyjmie on domyślnie nazwę QUEDEX-1987-THALAMUS-A-.TAP_C64 ROM-TAPE DATA_BLOCK_5. PRG. Program dodaje numer porządkowy według tego ile segmentów wykryje. Podczas opisu podawałem ich numery, więc nie mam możliwości pomyłki oraz je oznaczyłem następującymi kolorami: czerwonym – pliki muzyczne, zielonym - gry, niebieskim – obrazka oraz pomarańczowym – pliki pomocnicze zawierające informacje ułatwiające lokalizację poszczególnych etapów ładowania. Standardowy loader firmy Commodore zawiera dwukrotnie zapisane to samo – nagłówek (HEADER) oraz dane (DATA). Wyjaśnienia wymaga format PRG. 2 pierwsze bajty określają adres ładowania pliku a bezpośrednio za

nimi znajdują się dane. W tej części nie skorzystałem z opcji łączenia następujących po sobie segmentów. To tyle tytułem wstępu.

1. US-GOLD – Ten loader posiada bardzo grube kolorowe paski. Zobaczmy jak to zrobić na przykładzie gry Blasteroids. Po otwarciu pliku przechodzimy na PRG Extraction. Interesuje nas plik zawierający grę (segment 5). Zaznaczamy go, klikamy Export PRG i zapisujemy na dysku.



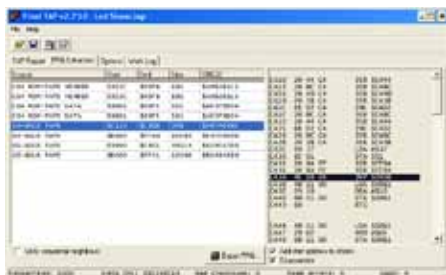
Jak widać na obrazku adres uruchamiający to \$0700. Wystarczy teraz spakować grę np.za pomocą programu exomizer i cieszyć się oryginałem.



Zastanawiacie się na pewno skąd wiedziałem jakim adresem uruchamia się ta gra. Wystarczy przeanalizować ten krótki kod z segmentu DATA. Na początku następuje ustawienie Timerów i wyłączenie ekranu, i odcodowanie programu od adresu \$0858. Dalej następuje przepisane części loadera do bufora magnetofonu oraz na stos. Pod adresem \$0179 następuje skok do kolejnego uruchamianego programu – czyli w tym przykładzie \$0700.

Nie należy się zawsze sugerować tym co pokazuje program gdyż twórcy czasami potrafili nas przechytrzyć, podając jako początek komórki 0 np. w grze Impossible Mission II.

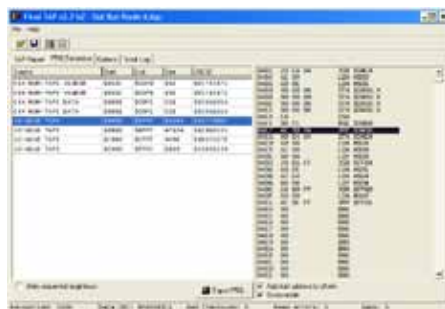
Istniały również wersje jednoczęściowych gier z obrazkiem. Led Storm jest dobrym tego przykładem. Po otwarciu obrazu kasety i przejściu na zakładkę PRG Extraction widzimy trochę więcej plików. Tu sprawa jest odrobinę zagmatwana. Po uruchomieniu programu następuje właściwe ładowanie gry. Pierwszy plik typu US-GOLD (segment 5), korzysta ponownie z loadera, który się wcześniej załadował, oraz program wyświetlający obrazek i uruchamiający grę. Kolejne pliki to obrazek (segment 6) i gra w dwóch częściach (segmenty 7 i 8).



Do składania gry należy zacząć od końca, czyli wyciągnąć 2 ostatnie pliki i je spakować. Jako adres uruchamiający należy podać \$0400. Teraz pora na dodanie obrazka. Jego początek znajduje się od \$d800. Kompilujemy kod listingu 1 i zapisujemy w pliku PRG. Teraz pakujemy ponownie 3 pliki – grę, program do wyświetlania obrazka oraz sam obrazek. Jako adres uruchomienia podajemy \$C000. Wszystkie potrzebne dla nas informacje znajdują się w 1 pliku US-GOLD (segment 5) od adresu \$c44d. Na listingu 1 przedstawiłem przykładowy program wyświetlający obrazek oraz uruchamiający wcześniej spakowaną grę.

Ostatnim przykładem będzie gra Outrun. Po otwarciu mamy 4 pliki US-GOLD. Pierwszy (segment 5) przepisuje obrazek, wyświetla go i wczytuje pozostałe 3 pliki z grą

(segmenty 6 - 8). Po załadowaniu 2 pliku US-GOLD (segment 6), ekran zostaje wyłączony i następuje załadowanie obszaru od \$c000 - \$ffff. Aby wyciągnąć samą grę należy zapisać 3 ostatnie pliki. Adres uruchamiający grę znajduje się pod \$949d co widać na poniższym obrazku. Jeśli kogoś interesuje obrazek musi zapisać dane od \$c000-c3e8, \$e000-ff40 oraz \$d800-dbe8 i odpowiednio pozmienić adres wyświetlanego ekranu. Można by się pokusić o zrobienie wyboru odpowiedniej trasy (Route A-E).



2. ATLANTIS TAPE – Nieskomplikowany loader zawierający wiele interesujących gier. Ja pokażę na przykładzie 3 gier jak można je zapisać. Jako pierwsza będzie gra Super kid. Po otwarciu pliku widać 4 pliki typu ATLANTIS TAPE. Pierwszy zawiera program wyświetlający obrazek, oraz uruchamiający grę (segment 1). Pozostałe to wspomniany obrazek (segment 2) oraz gra w 2 częściach (segmenty 3 i 4). Jak widać po adresach początku i końca 3 ostatnich plików, nie należą one do siebie, dzięki czemu można w łatwy sposób rozpakować grę i obrazek. Jak tego dokonać ilustruje listing 2. Zasada działania jest identyczna jak w przypadku gry Led Storm, z tym, że jako adres uruchomienia gry należy podać \$0900. Natomiast plik po rozpakowaniu uruchamiamy \$5200. Teraz wystarczy to spakować i zamierzony cel zostanie osiągnięty.



Pora na drugą grę – Cavemania. Z tą grą jest podobnie jak z poprzednią. Różnica to zmiana koloru przed petla1 z `lda #$0a` na `lda`



#\$01 z listingu 2. Resztę czynności wykonujemy jak poprzednio.

Ostatnio grą tego loadera będzie Montor. Wszystko wyglądałoby jak poprzednio gdyby adresy plików obrazka i gry nie pokrywały się z sobą. W takim przypadku należy umiejscowić go w innym miejscu. Ze względu, że nie ma wielu możliwości zostaje ostatnia ćwiartka pamięci. Zapisujemy 3 ostatnie pliki (segmenty 6, 7 i 8) na dysku i zmieniamy 2 bajt wyeksportowanego pliku obrazka z \$58 na \$d8. Teraz korzystamy z listingu 1 i odrobinnę go modyfikujemy. Jako początek podajemy \$cc00, adres uruchomienia gry to bez zmian - \$0900. Teraz pakujemy 4 pliki i uruchamiamy od \$cc00.

W przedstawionych 3 przykładach po ukazaniu się obrazka i odczekaniu 5-cio sekundowego opóźnienia gra uruchamia się od razu. Z tej reguły należy korzystać, gdy istnieje możliwość umieszczenia obrazka w niezajętym miejscu pamięci. W pozostałych przypadkach należy spakować najpierw grę, a później dane, które będą się wyświetlały w pierwszej kolejności.

3. WILDLOAD – Specyficzny i nieco skomplikowany loader. Tym razem oprócz suchych informacji, będę musiał nieco więcej wyjaśnić odnośnie uruchamiania oprogramowania. Przykładowo w demie gry Delta Mixtape po jego załadowaniu widzimy dokładnie to samo, co podczas ładowania gry Delta, w jej początkowej fazie. Po otwarciu pliku widać 2 pliki HEADER, 4 DATA oraz 4 interesujące nas WILDLOAD. Sprawa wygląda niemal identycznie jak wcześniej. Drugi i trzeci plik zawiera demo (segmenty 8 i 9), natomiast ostatni (segment 10) informację, gdzie należy go uruchomić. Znajduje się on w nim zawsze między \$02d0 - \$02e7. Wyeksportowane pliki pakujemy i podajemy adres uruchomienia jako \$ccdd.

Wydaje się, że to nie skomplikowane. Zobaczmy na przykładzie dema gry The Last Ninja. Tutaj jest jeszcze łatwiej, bo są tylko 3 pliki. Drugi to demo (segment 8) a trzeci (segment 9) zawiera jak i poprzednio potrzebne nam dane do jego uruchomienia. Jednak po podglądzie dis aseblacji widzimy takie coś.

```
02DB 20 59 A6 JSR $A659
02DE 20 33 A5 JSR $A533
02E1 4C AE A7 JMP $A7AE
```

Co to oznacza, nie ma żadnego konkretnego adresu. To są informacje potrzebne do wykonania uruchomienia poleceniem RUN. Procedura powoduje przywrócenie podstawowych informacji w komórkach określających początek i koniec pamięci, łańcuchów oraz zmiennych. Czyszczony jest także stos. Następnie następuje obliczenie szczytu końca programu w BASIC-u, wartość 0 oznacza koniec programu. Teraz należy pobrać kolejną linię programu i ją wykonać. Aby uruchomić to demo musimy umieścić gdzieś ten krótki

program. Ja umieściłem go od \$c000 co oznacza adres jaki należy podać po rozpakowaniu. Dodatkowo rozpakowany plik jest od \$0800, co powoduje, że nie musimy zerować tej komórki po rozpakowaniu. Jest to jakiś defekt tego pakera.

Zobaczmy teraz grę Crazy Coaster. Podczas jej ładowania można pograć w inną grę – Micro Painter. (segment 8) Po zakończeniu ładowania uruchamia się już wspomniana na początku gra. Zasada jest identyczna, czyli ostatni plik (segment 12) zawiera adres uruchomienia, a dwa wcześniejsze (segmenty 10 i 11), to gra. Jednak tym razem można także zapisać grę, wczytaną i uruchamianą na początku ładowania. W pierwszym lub drugim zdublowanym pliku DATA (segment 3 lub 4), na samym jego końcu znajduje się adres uruchomienia pod adresem \$02ec. Gra włączy się, gdy wartość komórki \$02 nie wynosi 0, co następuje po wczytaniu trzeciego pliku WILDLOAD. Mając wszystkie potrzebne informacje uruchomimy grę. Ze względu na jej położenie musimy wyłączyć przerwanie rozkazem SEL, następnie wyłączyć KERNAL zerując pierwszy bit w komórce \$01 i skoczyć do adresu \$e000. Po napisaniu programu umieszczamy go gdzieś w pamięci i dodając plik gry pakujemy je.



Niekiedy dodawano obrazki, jak choćby w grze Spin Dizzy. Jednak tym razem ma on długość tylko 4KB. Sprawa jest oczywista, jest wykonany ze znaków graficznych a nie grafiki. Zapisujemy 3 pliki: obrazek (segment 12), oraz dwa z grą (segmenty 14 i 15). Najpierw wyświetlamy obrazek, odczekujemy chwilę i uruchamiamy grę adresem \$6000. Plik obrazka też można skrócić usuwając od \$1002 (bez offsetu) wszystko do końca. Należy wyjaśnić jak wyświetlany jest obrazek ze znaków graficznych. Po ustaleniu banku wideo dla VIC-a, należy podać adres ekranu oraz kształtu znaków graficznych. Te informacje odczytywane są w komórce \$d018. Cztery starsze bity (7-4) oznaczają początek ekranu przesuniętego w KB i może on przyjmować wartości od 0 do 15. Bity 3 -1 oznaczają przesunięcie adresu kształtu znaków graficznych wyrażonego w KB i może on przyjmować wartości od 0 do 14. Jak widać na listingu 3 wartość \$a8 oznacza ustalenie adresu kształtu znaków graficznych przesuniętego o 8KB, czyli od \$e000 oraz ekranu o 10 KB - \$e800. Od \$eC00 znajduje się 1KB kolorów które są przepisane do pamięci kolorów VIC.

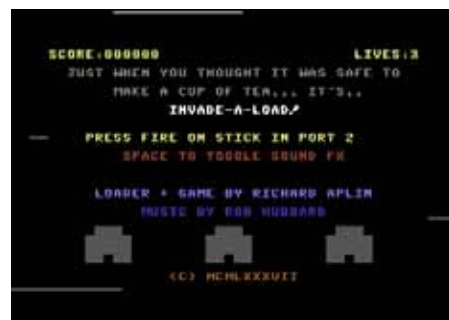


Na koniec opisu tego loadera przyjrzymy się grze QUEDEX. Tym razem plików jest mnóstwo. Po podglądzie ładowania widać obrazek z muzyką, później jest ona wyłączona a na ekranie pojawia się komunikat o demie zapisanym bezpośrednio po grze, po czym następuje uruchomienie samej gry. Zaczynamy znowu od spakowania gry, która znajduje się w 3 najdłuższych plikach (segmenty 17, 20 i 21) przed demem rozpoczynającym się od HEADER. Adres odczytujemy jak zwykle z krótkiego 89-cio bajtowego pliku, który wynosi \$8400. Teraz pora na dodanie obrazka i muzyki. Sama bitmapa ładuje się od \$e000, natomiast kolory i ekran tekstowy mają identyczny adres \$d800 i mają długość 1023 bajtów. W takim razie musimy zmienić adres ekranu na np. \$d000. W tym celu otwieramy pierwszy plik o tej długości (segment 14) w HEX edytorze i zmieniamy drugi bajt na \$d0. Teraz czas na muzykę. Tu sprawa jest ciekawa. Z miejsca jej uruchomienia podczas ładowania - \$cf00, co widać w pierwszym pliku DATA (segment 3 lub 4) i kod programu jest umieszczony w pliku WILDLOAD o adresie \$ac0 (segment 12) wynika, że wystarczy skakać do \$cf75 i odczekać jedno odświeżenie ekranu, aby tempo było prawidłowe. Jednak nic nie chce grać. Okazuje się, że wywołanie procedury następuje od \$c000, gdzie ustalany jest adres przerwania IRQ na \$c01f. Gdy wyeksportujemy i załadujemy tylko plik z muzyką i wykonamy SYS49152, to oczywiście usłyszymy grającą melodię. Jednak jej tempo jest nieco szybsze niż faktycznie. Aby grało prawidłowo musiałem przeanalizować kawałek kodu, i okazało się, iż za każdym razem program zmienia komórkę \$01 na \$36 w celu odczytu danych z RAM -u a nie BASIC-a. Tera już wszystko powinno być już zrozumiałe. Wystarczy skopiować część kodu od \$cf00 do odtwarzania melodii, a przed jej wywołaniem odjąć 1 od komórki \$01. Dodatkowo można usunąć wszystkie dane od \$2050 (bez offsetu) w HEX edytorze z pliku zawierającą melodię. Na listingu 4 jest to przedstawione.



Posiadając już wystarczającą wiedzę możesz w ten sam sposób zapisać demo QUEDEX, znajdujące się w pliku bezpośrednio za grą.

4. FREELoad – Loader stosowany przez wiele firm umożliwia pogranie w grę Space Invaders podczas wczytywania. W każdym nagraniu tego typu należy wyeksportować ostatni plik. Powinien zaczynać się on od \$0801, co występuje w wielu grach. Gra umieszczona jest tylko wtedy, gdy ładowane po niej dane nie osiągają adresu \$7000. Wyeksportowane gry nie są spakowane, więc można zmniejszyć ich wielkość kompresując je. Korzystały z niego m. In. takie gry jak Ultimate Combat Mission,, Angleball, Uridium i wiele innych. W przypadku ostatniej gry adres uruchomienia to \$0900 co dokładnie widać w programie. Aby wyciągnąć grę Space Invaders zapisujemy 3 pierwsze pliki (segmenty 5 - 7) i dodajemy nieskomplikowany listing 5.



Na koniec coś trudniejszego. Okazuje się, że ten loader tylko z nazwy ma coś wspólnego z innymi grami, które korzystały z niego. Przykładowo w grze Puzznic wygląda to dość specyficznie. Na podglądzie można zauważyć powtarzający się plik obsługujący przesuwanie napisu, granie melodii czy wyświetlający obrazek (segmenty 5, 7 i 16).



Pierwszymi interesującym nas segmentami są te zawierające obrazek (segmenty

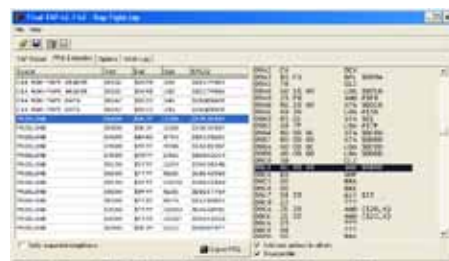
9 - 11). Należy zmienić adres ładowania segmentu 9 na \$d800 i 10 na \$dc00. Natomiast ostatnie pliki (segmenty 12 - 14, 17 i 18) to gra. Najpierw pakujemy grę, a następnie korzystając z listingu 1 oraz obrazka łączymy wszystko w całość. Adres uruchomienia obrazka jest taki jak na listingu 1, natomiast gry to \$0400. Skąd wiadomo że akurat taki. Podczas ładowania przepisywany jest program na stos (\$09e4 - \$0a45) pod adres \$0112. Pod adresem \$0170 lub \$0a42 umieszczony jest JMP \$0400 – czyli skok do samej gry. Jak już wspomniałem loader odgrywa także muzyczkę. Nie dodałem jej do gry, gdyż nie jest ona jej integralną częścią. Można ją także posłuchać przy wczytywaniu gry Cabal.



Okazuje się jednak, że oprócz tak wyglądającego sposobu ładowania, zastosowano zupełnie inaczej wyglądający w grze Slap Fight. Jest on nieco mniej skomplikowany od poprzednika, mimo iż sam listing 6 jak i związany z nim kod, jest bardziej obszerny. Tym razem musimy skorzystać z wszystkich plików oznaczonych jako FREELoad. Zajmują one znacznie więcej pamięci, niż jest jej dostępnej. Tak więc tradycyjnie zaczynamy od samej gry. Tym razem spotkała mnie niemiła sytuacja. Po podglądzie sposobu ładowania (muzyka i rysująca się grafika), wynika ze następująco bezpośrednio po niej pliki są samą grą. Po spakowaniu gra się uruchomiła, jednak bez sprajtów. Więc zacząłem od początku analizować kod (segment 5 lub 6) i okazało się, że pierwszy plik wczytywany pod adresem \$4000 to plik gry. Jest on przepisywany do \$d000, teraz wszystko stało się jasne. Znajduje się to od adresu \$08aa-\$08c7. Spakowałem grę (segmenty 8, 12 - 17) zmieniając adres tego pierwszego na podany przed chwilą. Teraz pora na muzykę (segment 7) i grafikę (segmenty 5 i 9 - 11). Teraz nastąpiło u Ciebie ogólne zdziwienie, czemu segment 5 jest oznaczony jako plik zawierający przydatne informacje oraz grafikę? Zawiera on tabelkę potrzebną do wyświetlania sprajtów. Zaczynając od muzyki musimy ją zainicjować wywołując podprogram o adresach \$a32f, załadować do rejestru Y 5 i wywołać \$a23d. Teraz pora na nieskończone granie muzyki. Odczekując zawsze do tej samej linii rastra - \$d012, wywołujemy to co znajduje się od etykiety prog. Jest tam kilka skoków JSR. Mając już to opanowane pora na obrazek. Jest on rysowany czyli wpisywane są odpowiednie wartości do kolejnych komórek. To ilustruje podprogram wyświetl. Zastosowałem przed jego wywołaniem trzy krotne przyspieszenie, aby efekt był wizualnie zbliżony do tego z kasyty. Po przepisaniu ca-

łego obrazka następuje inicjacja sprajtów oraz ich animacja. Aby zmieścić się, obrazek umieściłem od \$80c0 skracając plik do \$1f42. Mapa kolorów (segment 10) umieściłem od \$dc00 natomiast ekran z dodatkowymi kolorami dałem od \$c000. Ten plik zajmuje trochę więcej niż 1KB, bo zawiera sprajty. Poskładajmy to w całość. Najpierw kasujemy 8 ostatnich KB RAMu, przepiszujemy dodatkowe kolory od \$c000 do \$d800, oraz kopiujemy kolory ekranu od \$dc00 do \$c000. Ustawiamy grafikę multikolor i włączamy ją. Uruchamiamy muzykę oraz rysujemy grafikę przepisując po 3 bajty na jedno odświeżenie ekranu. Po wypełnieniu całego ekranu inicjujemy sprajty – sprite_init (\$0a2b) oraz uruchamiamy animację – spr_wysw (\$0a7e). Z segmentu 5 kopiujemy dane od \$09c4 do \$0a03 i jako początek ładowania podajemy \$c9c4. Do przerywania odgrywania muzyki naciskamy spację. Sprawdzenie użyjemy porównując komórkę \$dc01 oraz bit 4, i jeśli tylko on przyjmie wartość 0 nastąpi rozpakowanie gry. Adres jej uruchomienia znajduje się pod \$09c1 i wynosi \$0600.. Pakujemy wszystko dodając listing 6 i uruchamiamy od \$cd00.

Przedstawione przykłady jedynie nakreślają sposób obchodzenia się z loaderami. Do wyczerpania tematu każdego z nich musiałbym podać znacznie więcej przykładów. Nie ma jednoznacznego schematu postępowania i zawsze trzeba improwizować.



W dodatkach do tego numeru znajdziesz opisane listingi jak i sam program Final Tap. Tabela przedstawia poziom kompresji poszczególnych plików. Na YT możesz obejrzeć rezultaty moich poczynąń.

<http://www.youtube.com/watch?v=eSod-6Fs1HgM> oraz <http://www.youtube.com/watch?v=jyTLVXsKwnM>

11111olo
c64.com.pl@gmail.com

Tytuł	Loader	Rozmiar	
		Przed	Po
Angleball	FREELoad	17758	11600
Blasteroids	US-GOLD	63744	31277
Cavemania	ATLANTIS LOADER	39736	20714
Crazy Coaster	WILDLoad	43010	20250
Delta Mixtape	WILDLoad	11493	5834
Invade A Load	FREELoad	7951	4689
Led Storm całość	US-GOLD	47265	41651
Led Storm gra	US-GOLD	61256	37104
Micro Painter	WILDLoad	8200	3571
Moontorc	ATLANTIS LOADER	49851	26614
Out Run (Route A)	US-GOLD	59391	27706
Puzznic całość	FREELoad	33702	26915
Puzznic gra	FREELoad	60411	23548
QueDex całość	WILDLoad	50570	40243
Que-Dex gra	WILDLoad	63488	31924
Que-Dex demo	WILDLoad	18517	13057
Slap Fight całość	FREELoad	49574	40563
Slap Fight gra	FREELoad	64497	30876
Spin Dizzy	WILDLoad	47468	26671
Super Kid	ATLANTIS LOADER	39036	19514
The Last Ninja Demo	WILDLoad	41203	26200
Ultimate Combat Mission	FREELoad	22168	15781
Uridium	FREELoad	46594	25446

Róbmymy Grę.



Muszę wam się do czegoś przyznać. Sztuki programowania na C64 uczyłem się nie dlatego, żeby pisać wspaniałe prezentacje typu demo, szokować widzów niesamowitymi efektami, ścigać się z najlepszymi koderami i być poważanym w środowisku scenowców, ale po to by robić gry...

Tak, czułem głód dobrych pozycji na rynku gier, – ale dlaczego na Commodore 64, przecież to był najlepiej oprogramowany (czyt. miał najwięcej gier) komputer na świecie! Zgadza się, ale nie oszukujmy się, zdecydowana większość z tych produkcji była kiepskiej jakości. Może i zachwycały w czasach, gdy „Sześćdziesiąt cztery” zaczynał karierę, na tle innych platform z tego okresu, ale ja patrzyłem jednak na to z perspektywy polskiego entuzjasty z początku lat dziewięćdziesiątych – kiedy to właśnie dotarł do nas na dobre. Wtedy to PC-ty stały na wystawach gdzieś na targach, nie było jeszcze internetu, a i Amiga dopiero raczkowała, o Apple nawet nie słyszałem, Pegasusa i świata konsol jeszcze „nie przyciągnęli” ze wschodu. W domach królowały ośmiobitowe Spektrum/Timex, Atari i Commodore. W zasadzie poziom gier wśród tej trójcy był całkiem wyrównany – a różnice przeważające na korzyść którejś z platform wyznaczało kilka pikseli więcej na bohaterze lub użycie innego koloru.

Jednak były takie miejsca, gdzie gry nabierały nowego wymiaru – mowa o tu o „Salonach Gier” (pamiętacie?). Czasem stałem godzinami i patrzyłem na te olśniewające graficznie, dopracowane animacje i efekty dźwiękowe, jakże z wielką „swobodą” toczyła się tam akcja. Wiele z tamtych tytułów można było odnaleźć w tej przebogatej bibliotece C64, ale równie szybko okazywało się, że podobieństwa były, co najwyżej średnie, żeby nie powiedzieć czasem niezauważalne – zdarzały się również rodzyнки, ale w zdecydowanie śladowych ilościach.

Oczywistą sprawą było też to, że ówczesne automaty do gier były już bardziej zaawansowane technologicznie niż nasz pocztowy C64 i na początku tak sobie tłumaczyłem te różnice w wersjach. Jednak z czasem zauważyłem, że niektóre gry mimo słabszych walorów graficznych i muzycznych potrafią również przykuć do joysticka na wiele godzin tak jak te z „Salonu”, ba... czasem były lepsze. Gdzie

tkwi haczyk?

W sumie jest to dość trudna do określenia właściwość słowna, po angielsku brzmi to „gameplay” po polsku było kilka prób podejścia: „atrakcyjność”, „grywalność”, „okejkowość”, „miodność” itd. Po prostu te wersje z C64 gdzieś to gubiły. Co dokładnie o tym decyduje? Grafika i muzyka? Też, ale na pewno samo to nie wystarczy (najwyżej na screenshot do gazety) – musi być opracowana dobra metoda sterowania bohaterem i całą akcją w grze. Inaczej każde „wodotryski” szybko wyblakną. Tego mi właśnie w wersjach na C64 często brakowało, a nawet niektóre pozycje były dla mnie zupełnie „niegrywalne” i zastawiałem się, po co je ktoś stworzył.

Cóż, nie pozostało nic innego jak brać się za robotę. No i zacząłem tworzyć własne wersje – najpierw w Basicu, potem po trochu dokładałem elementy kodu maszynowego, aby w końcu zupełnie przejść na programowanie w assemblerze.

Projekty często kończyły się już na pierwszym etapie, czyli stworzeniu jakiejś „graficzki” tła i duszków i próbowaniu nadać im trochę życia w postaci ruchu. Niestety im dalej człowiek zagłębiał się w tajniki tworzenia komputerowej rozrywki tym bardziej okazywało się to trudniejszym przedsięwzięciem, ilość barier natury technicznej i problemów algorytmicznych do rozwiązania rosła wręcz w geometrycznym tempie przy każdym kolejnym kroku. O zgrozo, zupełnie nie widać na ekranie gry, tego co musi się dziać w „środku” aby wszystko trzymało się kupy, nie mówiąc o tym żeby jeszcze było to efektowne. Co tu owijać w bawełnę pisanie przyzwoitych gier wcale nie jest proste.

Po latach, nadal nie doczekałem się nawet kilku procent realizacji swoich growych marzeń, ale za to przybyło trochę doświadcze-



Bomberman c64 w swojej krasie

nia, którym chciałbym się z wami podzielić.

Od razu się pochwalę, że udało mi się napisać jedną z ciekawszych współczesnych pozycji na komcia w grach – mowa tu oczywiście o Bombermanie C64.

Nie był to oczywiście w żaden sposób produkt oryginalny, bazował bowiem na dość znanej grze z innych platform o tej samej nazwie – człon C64 dodałem, żeby wszystko było wiadomo :-), lub Dynablast (wersja europejska), ale nie korzystałem z żadnych źródeł oryginału, a cały kod gry pisany był tylko przy pomocy jej obserwacji (a właściwie kilku jej wersji) z roli gracza. Co prawda Bombermana C64 pisałem sam (świetną muzykę dorzucił od siebie Conrad), ale historia tego projektu sięga jeszcze połowy lat 90-tych, kiedy nad grą pracowaliśmy w grupie kilku osób – tu pozdrowienia dla Rasputina, Loba i Gutteda. Niestety tamten projekt nie doczekał się ukończenia, chociaż jakimś sposobem „wyciekł” gdzieś do sieci – można go znaleźć pod tytułem „Boombastic Benny”. Oczywiście, praca w grupie, zdecydowanie szybciej posuwa takie projekty do przodu, jest więcej pomysłów i entuzjazmu, ale to oczywiste. Niestety okres w którym kończyliśmy projekt to zmierzch ery 8-bitowej w Polsce, a i czas rozpoczęcia nauki na szkołach wyższych. Boombastic Benny umarł „śmiercią naturalną” zanim się urodził. Po latach jednak zapragnąłem wrócić do tego tytułu i dokończyć dzieło, jednak szybko okazało się, że: raz – musiałem przypominać sobie programowanie, dwa – człowiek odwykł do korzystania z edytora Turboassemblera (a już od commodorowskiej klawiatury to w ogóle), trzy – stare źródła projektu były w takim „nieładzie” na dyskach 5 i 1/4 cala, że łatwiej było zacząć wszystko od nowa... Trochę samozaparcia, przy okazji przejście na crossassembler dały w rezultacie pozycję, która była lepsza od poprzedniczki, i doceniona w obecnym świecie c64, zebrała nawet kilka pochwał i nagród – a liczba downloadów z csdb tej pozycji sięgnie niedługo 3 tys. Świadczy to o całkiem lubianym produkcie – to cieszy.

No dobra, już się pochwaliłem, ale teraz do rzeczy i czas na odwieczne pytanie – od czego zacząć pisanie gry? Pomysł – oczywiście – oryginalny zdecydowanie lepszy, ale wśród tylu tytułów i rodzajów jakie są dostępne obecnie na rynku, jest to bardzo trudne, i nieoryginalny też może być :) Powiem tak, im więcej spraw „przewidzimy” na początku, tym później łatwiej i szybciej nam będzie to wdrożyć do kodu, a jak będzie to pozycja już dostępna np. na innej platformie to już „kawałek roboty” mamy załatwiony. Oczywiście nie zamierzam tylko teoretyzować i równolegle przedstawię przykładowy projekt pewnej gry, który nawet

przygotowałem pod potrzeby tego artykułu. Na jego podstawie postaram się wyjaśniać i zwracać uwagę na problemy i sposoby rozwiązań wybranych zagadnień. Oczekuję też żywej dyskusji, która może nadać kształt możliwie, że kolejnym częścią takiego kursu. Na początek niewiele, ale od czegoś trzeba zacząć. Gra jest prosta „jak budowa cepa” (hehe, tak, wszyscy zapewne wiedzą, że cep składa się z dzierzaka, gązwy i bijaka), bo nie zamierzałem tworzyć skomplikowanych projektów.



Szczerebiec w akcji

Mamy bohatera, który ma miecz i ma robić z niego użytek – proste nie? Ile taka gra może zajmować pamięci? 1kB, 2? Nie. Rozpakowane dane grafiki w wersji preview już zajmują ok. 200 kB! Mało tego, jak zacząłem pisać kod, okazało się, że rozrósł się do kilku tysięcy linijek w kilkunastu plikach. No niestety, prymitywna grafika i dźwięki, która towarzyszy grze, nie oznacza wcale, że program też jest prymitywny. Po co się męczyć w takim razie i pisać taką „nieładną” grę? Bo, jest fajna – niestety, nie jest to mój autorski pomysł, i przy pisaniu oparłem na „sprawdzonym” pomysle, jak w przypadku wyżej opisywanego Bombermana. Jakiś czas temu w sieci natrafiłem na tzw. „flashową gierkę” o nazwie „Ginormo Sword”. Już od pierwszego momentu urzekła mnie swoją pisklowatością i prymitywną grafiką, ale to był dopiero początek. Ta prostota to był właśnie klucz do sukcesu – kilka tygodni później, pokonując ostatniego „Boss’a”, sam już nie wiedziałem ile spędziłem przy niej godzin, ileset tysięcy wykonałem kliknięć myszką, aby zabijać niezliczone hordy kreatur znanych z fantastyki. Stwierdziłem również, że nie jestem jedyny, który z taką pasją zainteresował się tą grą, i jest cała masa graczy, która również tę pozycję okrzyknęła jako kultową. No ale zaraz, przecież co w tej grze jest kultowego? Grafika jak z ośmiobitowców i pomysł jak z lat 80-tych. Hmm wiecie co, ja nie mogłem odnaleźć ani jednej takiej pozycji w „przebogatej” bibliotece na c64, i na innych platformach 8-bit również. Jak to możliwe? Gra, która wydaje się właśnie kwintesencją retro 8bit, nie ma odpowiednika na tego typu platformach. Jak się w nią zagłębimy to okaże się, że wcale nie jest taka prosta do wykonania na takich komputerach.



Ginormo Sword - protoplastuś

Oczywiście jej prostota jest tylko na pierwszy rzut oka, w rzeczywistości, w trakcie rozgrywki gra się rozbudowuje. I tu mamy do czynienia z podstawowym zabiegiem podnoszącym atrakcyjność produkcji – start ma być prosty i niezobowiązujący, dalej trudność ma stopniowo wzrastać, ale mają też przybywać nowe elementy. W ten sposób łatwo zacząć i się nie zniechęcić, a potem nie znudzić! A więc gra ma sporą liczbę zróżnicowanych przeciwników, którzy zapełniają odpowiednią bibliotekę zdobywcy. Posiada również kilka różnych krainek, z lokacjami bitewnymi, sklepikami, świątyniami, gracz rozbudowuje swoje zdolności, czary itd. Tak, to gra typu RPG, ale nie nachalna. Nie zarzuca już od początku ekranem pełnym różnych opcji i nie dzieli go na różnego rodzaju sekcje dla inwentarza, czarów, członków załogi, uzbrojenia itp. To wszystko tu, pojawia się dopiero w odpowiednim czasie i w najprostszy sposób, w końcu powstała już cała masa gier tego gatunku, z których wzorce można czerpać garściami. To dlatego nie możemy znaleźć tej pozycji w gronie gier na 8bit (a właściwie również na 16bit). Bo jest to wyciąg kwintesencji z tego typu gier do dnia obecnego, i ich maksymalnego uproszczenia – zostaje sama śmietanka. Prosta grafika powoduje niezobowiązujące odczucia (to pułapka!), przystępujemy do rozgrywki zrelaksowani – a dla nas – commodorowców - oznacza to, że nie musimy ograniczać się parametrami sprzętu aby próbować dokonywać konwersji. Mało tego, autor jakby specjalnie ustalał ograniczenia z jakimi się borykają układy grafiki 8bit. A więc każda z kreatur ma maksymalnie 4 kolory, z niewielkiej palety barw, może być ich ograniczona liczba

na planszy (sprites), ba nawet dorzucił ramkę wokół obrazu jak na C64. Czyżby wzorował się na ograniczeniach właśnie VIC-a, a może sam kiedyś lubił „ciupać” w komodorowskie gry? Jak się później okaże, te ograniczenia to nie te z C64, bliskie, ale jednak trzeba będzie komcia chwilami napocić żeby je wykonać. Np. owszem mamy 4-ro kolorowe sprites, ale tu 2 są wspólne dla wszystkich wyświetlanych, a 4-ty to kolor tła (przezroczystość), a więc w porównaniu są 3 i to jeszcze z ograniczeniami.

Dzisiejszy odcinek traktuję jeszcze jako wstęp, i nie będę jeszcze was katował kodowaniem. Na razie proponuję zapoznać się z tym previewem gry - są wersje w dwóch językach „Szczerebiec” (nie zaszkodzi trochę historycznie), oraz po angielsku „Jagged Sword” - i poobserwować dokładnie co się tam dzieje, radzę również popatrzeć na oryginał w sieci i nawet porównać. W chwili gdy piszę ten artykuł, wersja demonstracyjna tej gry jest już umieszczona na portalu sceny CSDB i... już zaczyna się podobać, a od razu zastrzegam, że nie zamierzam samemu ją kończyć – to już będzie należało do was.

Wiem również, że nie tylko ja mam podobne ambicje pisania gier, a więc taki cykl artykułów może być pomocnikiem, wskazówką a może i czasem rozwiązaniem. Mam nadzieję, że skutecznie przyspieszy proces poznawczy zagadnienia tworzenia gier na c64, co w rezultacie spowoduje wysyp nowych pozycji, dla tego wdzięcznego przecież komputerka - a jest co pisać, no i jeszcze więcej jest wygłodniałych graczy. Bo co tu dywagować, magnesem jeszcze czasem ciągnącym ludzi do C64 są właśnie gry i z nimi jest on głównie kojarzony. To nara.

Skull(FanCA/Samar)

```
include „\C64Project\!sources\szzerebiec\inc\szzerebiec_player.asm”
include „\C64Project\!sources\szzerebiec\inc\szzerebiec_mob.asm”
include „\C64Project\!sources\szzerebiec\inc\szzerebiec_shot.asm”
include „\C64Project\!sources\szzerebiec\inc\szzerebiec_item.asm”
include „\CE4Pr0j&ct\!sources\szzerebiec\inc\szzerebiec_location.asm”
include „\C64Project\!sources\szzerebiec\inc\szzerebiec_events.asm”
include „\CE4Pr0j&ct\!sources\szzerebiec\inc\szzerebiec_routines.asm”

;file contain: a sets of packed gif binaries tables, etc
include „\C64Project\!sources\szzerebiec\inc\szzerebiec_data.asm”

-----
GAME_INIT
lda Location_Uncover
jsr Location_Set+3
-----
GAME_MAIN_LOOP
jsr Next_Frame
jsr Shot_service
jsr Player_service
jsr Mob_Service
jsr Events_service
jmp GAME_MAIN_LOOP
```

Kod głównej pętli gry

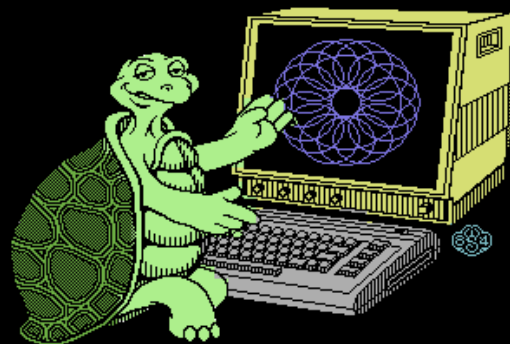
Hi-res

W Commodore 64 za układ graficzny odpowiedzialny jest VIC-II, który razem z chipem muzycznym SID był odpowiedzialny za sukces rynkowy tego komputera. Układ ten mógł wyświetlać tryb tekstowy 40 kolumn na 25 wierszy, a także dwa tryby graficzne. Pierwszy to hi-res (tryb wysokiej rozdzielczości), który oferuje rozdzielczość 320x200 pikseli z 1 bitem na piksel z możliwością rysowania dwoma kolorami z palety 16 kolorów. Jeden to kolor tła, a drugi to kolor rysowanego obiektu. Choć sam tryb ten ma duże ograniczenia to obecnie na demoscenie cieszy się dużą popularnością. Oto przykładowe obrazki w tym trybie rysowane w różnych latach C64.



Wayne Schmidt

Celtic Eagle - 1983



Wayne Schmidt

Turtle - 1984



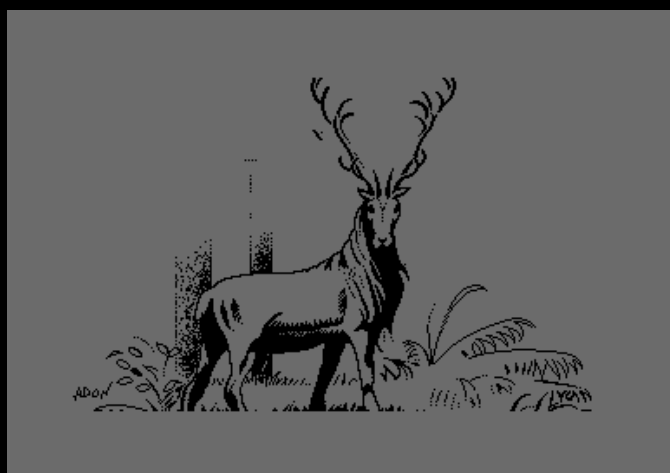
Wayne Schmidt

Irish Spring - 1984



Wayne Schmidt

Natalie - 1984



Adon

Jeleń - 1997



Zyga

Demon- 1997



Seba

Arashi - 1997



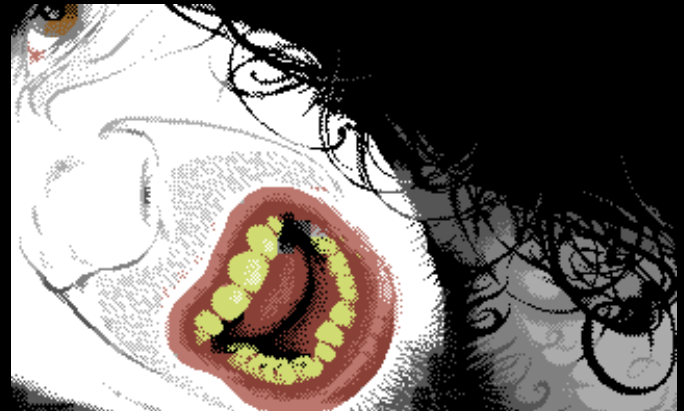
Gollum

Castle - 1997



Cruzer

Dexter64 - 1997



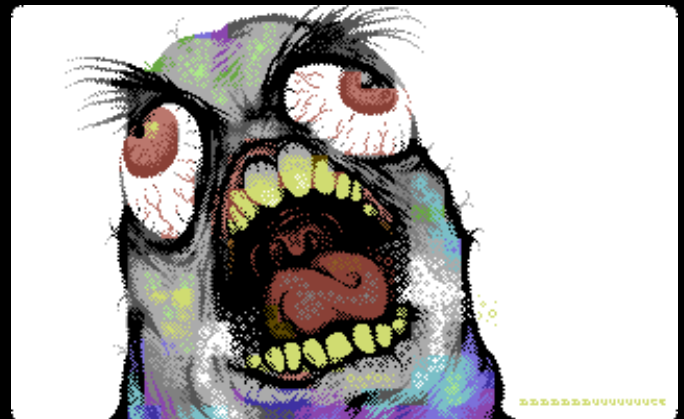
Duce

Selfish Shellfish - 2009



Duce

Must Ache the Moustache - 2010



Duce

TTTTTTUUUUUUKSU - 2010



Veto

The Mill - 2011



Duce

Tasmanian Dummy's Hangover - 2012