

ATARI - COMMODORE - ZX SPECTRUM - AMSTRAD CPC - AMIGA

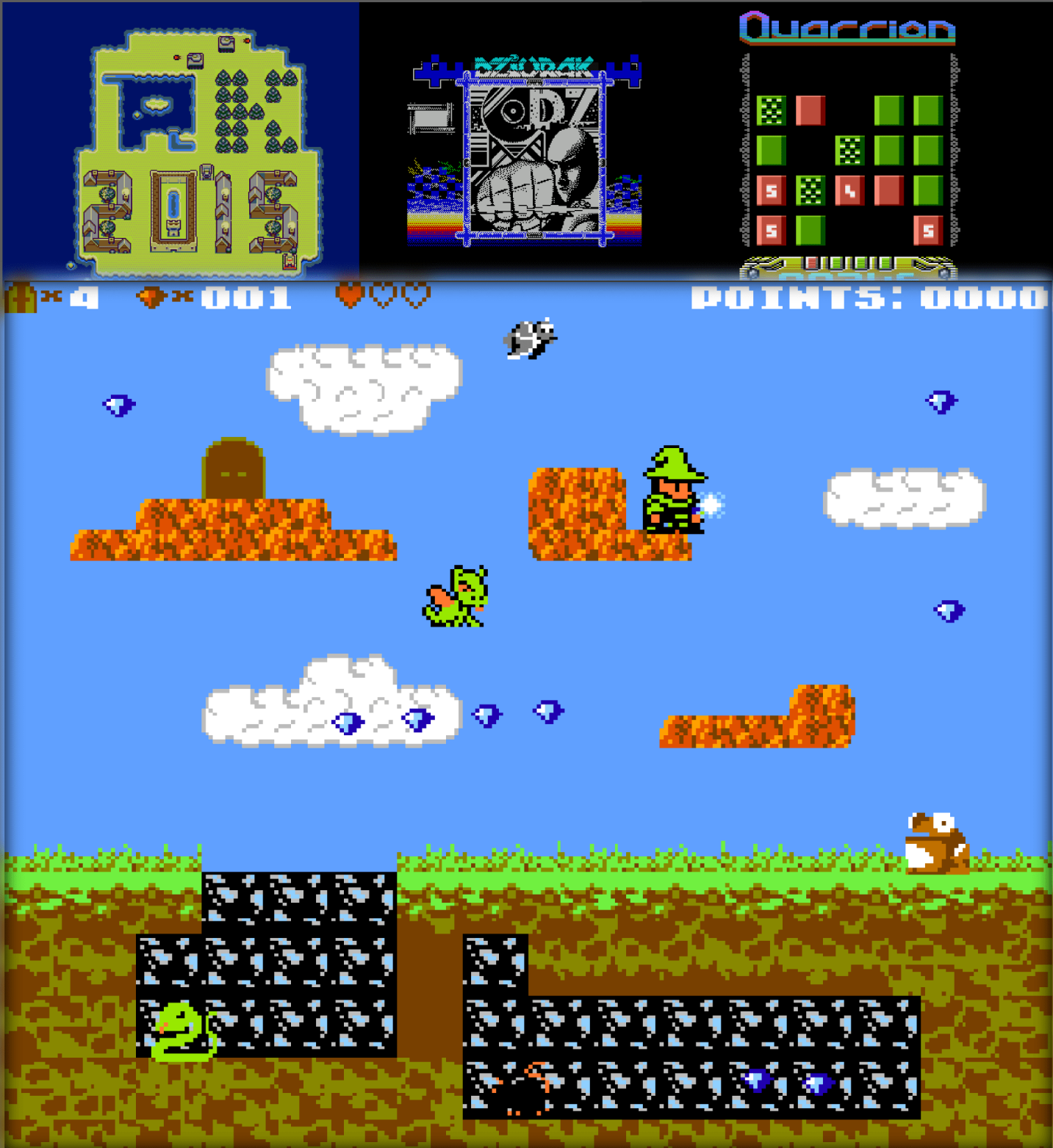
RETROKOMP

Mikrokomputery wczoraj i dziś

ISSN 2450-5862



9 772450 586007 >





Krzywa rośnie

Kiedy lata temu planowałem zakup swojego pierwszego komputera, nie przypuszczałem jak bardzo uzależnieni będziemy od technologicznego pędu w przyszłość. Nigdy bym nie pomyślał, że maszyna o mocy obliczeniowej określanej dumnym mianem "super komputera" będzie kiedyś stała na moim biurku. Co więcej, nie spodziewałem się, że nie będę chciał korzystać z wytworów techniki ery kosmicznej tak chętnie, jak ze starego dobrego ZX Spectrum czy Commodore.

Wiele myśli chodziło mi po głowie, ale na pewno nie to, iż 30 lat później nadal będę zajmował się tym samym sprzętem, na którym poznałem Automanię, Trap Door, Stealth czy Defender of the Crown. A gdzie ich następcy? Oczywiście powiecie - jest ich cała masa. Jasne, że rynek wyprodukował tysiące gier, z którymi - technicznie rzecz biorąc - stare tytuły nie mogą się równać. Coś jednak sprawia, że chcemy uruchamiać gry z datą premiery sprzed 20 lub 30 lat. I to nie jest czysty sentymentalizm.

Ile osób na świecie, tyle powodów zajmowania się tematyką retro. Z pewnością wielu z nas chce powrócić do beztroskiego czasu dzieciństwa lub młodości, gdy nie mieliśmy na głowie wielu problemów, a komputer był skrzynką o nie do końca poznanych możliwościach. Taki romantyczny obraz nie jest jednak pełną odpowiedzią. Wielu osobom, w tym mnie osobiście, nie podoba się kierunek, w którym postępuje rozwój świata IT. I to wcale nie dlatego, że ich ulubione Atari czy Amigi nie zdominowały rynku.

Patrząc na problem bardziej kompleksowo zauważymy szereg konkretnych czynników, które sprawiły, że w latach '80-tych zainteresowaliśmy się komputerami. Nie było przecież wtedy takiej

konieczności. Z pewnością nie odczuwaliśmy tak silnej presji powstających monopolii, bo rynek charakteryzował się większą równowagą. Istotą nauki informatyki było programowanie, bo nie było tak wielu gotowych rozwiązań na wyciągnięcie ręki. Obsługa sprzętu była z jednej strony zagadką, a więc doskonałą rozrywką intelektualną, a z drugiej - wejściem w nowy, wspaniały świat XXI wieku. Co zostało nam z tego dzisiaj? To, co sami będziemy chcieli zachować dla siebie i przyszłych pokoleń.

Dlatego w naszym magazynie piszemy o nietypowych rozwiązaniach problemów, które w swojej istocie mogą wydawać się proste. Gdy się nad nimi pochylimy, szybko okaże się, że nie są tak oczywiste jak nam się zdawało, a do tego wymagają nabycia szerokiej wiedzy. Nie odrzucamy zupełnie zdobyczy ostatnich lat, dlatego nie mamy nic przeciwko podłączaniu do retro sprzętu monitorów LCD czy też kart pamięci zamiast tradycyjnych stacji dyskiek. Za sprawą odpowiednich ludzi, możliwości naszych komputerów stale rosną i mogą one cieszyć kolejne lata.

Nie mniej istotne jest dzielenie się wiedzą i doświadczeniem z innymi użytkownikami, co można zrobić na jednym z wielu spotkań odbywających się w Polsce i na świecie. Jest ich dzisiaj niewiarygodnie dużo, a udział w największych biorą już nawet gwiazdy estrady. Możemy więc pokazać, że nasze zainteresowanie przekłada się na konkretną aktywność, a przy okazji zdobyć jedną z nagród, która ucieszy jeśli nie nas, to nasze pociechy. Pamiętajmy, że niekoniecznie trzeba od razu napisać grę czy demo, wszystko w swoim czasie. Warto jednak poczuć smak informatyki sprzed masowej komercjalizacji. Retro ma tak wiele twarzy, że każdy znajdzie coś dla siebie.

Adam Zalepa

Z listów do Redakcji:

(...) Jako użytkownik niezbyt popularnego komputera Acorn Archimedes, chciałbym spytać - czy będziecie pisać o mniej znanych komputerach? Rozumiem, że Waszym targetem są popularne platformy jak Commodore czy Atari, ale przydałoby się także parę słów o mniej znanych, a wcale nie mniej lubianych. (...)

Chcemy pisać o wszystkich ciekawych, również "nietypowych" komputerach. Oczywiście wiele zależy od bazy sprzętowej jaką dysponują autorzy artykułów. Nie chcemy powtarzać informacji jakie można przeczytać na stronach internetowych, z drugiej strony dotarcie do osób zajmujących się "niszą w niszy" nie zawsze jest możliwe. Z pewnością będziemy starali się zwracać uwagę na sprzęty, których jest w naszym kraju zdecydowanie mniej niż powyżej wymienione, popularne komputery.

RETROKOMP

Mikrokomputery wczoraj i dziś

NUMER 1/2016
(luty)

ISSN 2450-5862

Redakcja:
Adam Zalepa

E-mail:
retrokomp@amiga.net.pl

Autorzy tekstów:
Mateusz "Tfardy" Eckert
Adam Kłobukowski
Piotr „Piter” Krużycki
Piotr "Kroll" Mietniowski
Tomasz "Slayer" Pacyna
Michał "stRing" Radecki-Mikulicz
Krzysztof "Radzik" Radzikowski
Paweł "V-12" Ruczo
Piotr "Sachy" Sachanowicz
Rafał Szyja
Krystian "Tygrys" Włosek

Korekta:
Renata Gralak

Skład i okładka:
Andrzej Wilczyński

Wydawca:
"A2" Aleksandra Zalepa, Łódź

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



RETROKOMP 1 :

Krzywa rośnie
Doniesienia

WYWIAD 8

Wybór dla hobbysty

ZX SPECTRUM 12

Tryby graficzne Timexa
BASin - wygodny edytor i emulator ZX BASIC

AMSTRAD CPC 19

Metody konwersji sygnału RGB

ATARI 24

Interfejs SatanDisk
Atari Lynx - modyfikacja LCD i wyjście VGA
HDMI w służbie retro
FPS-y na mniejszego kota
Polskie znaki w systemach TOS, MiNT i Magic
Calamus 2015 - nowe podejście?
Adaptory kart graficznych do MegaST i TT
TenoxVGA - adapter do TT-ki
uIP - łatwiej się nie da
Wykopaliska

COMMODORE 40

Hot Style - dyskietka do poduszki
1581

AMIGA 42

Amiga Hardware Programming - wprowadzenie i odcinki 1-3
Amiga 600 - powrót do przyszłości?

INTEGRACJA 59

RetroKomp / LOAD ERROR 2015 - Raport z Party
Gamedev Compo
Silly Venture2k16 - Have you played ATARI today?



FireBee - ruszyła produkcja

Jest to ostatni moment, aby zamówić najnowszego kłona dużego Atari - FireBee - ruszyła właśnie produkcja drugiej edycji. Więcej informacji na:

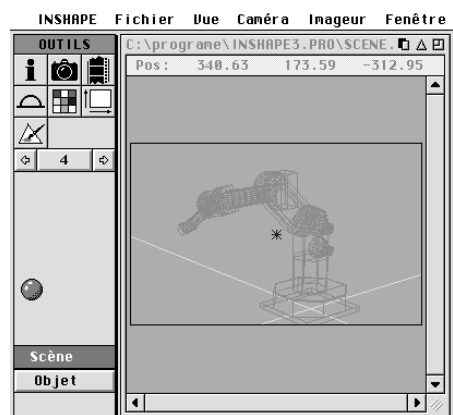
www.firebee.org

Inshape 3 na Firebee

Już za chwilę InShape 3 będzie przekompilowany pod Coldfire - czyli będzie działał na klonie FireBee. Program dostępny jest na razie na komputery Atari Falcon 030/060, zamówić go można na stronie:

inshape.lutece.net

a kosztuje 30 euro (płyta CD oraz instrukcja).



Początek roku i dwa, a może nawet trzy polskie atarowskie party w przygotowaniu!

W ostatni weekend maja (27-29.05.2016) odbędzie się kolejna edycja WAP-niaka, warszawskiego party poświęconego dwóm platformom: 8-bitowym komputerom Atari i ZX Spectrum. Pałeczkę organizatora po Sikorze przejął w tym roku Michał „Miker” Szpilowski. Zapraszamy, a więcej informacji znajdziecie pod adresem:

wapniak.atari.org.pl

Dwa razy SillyVenture w 2016 roku?! Prawdopodobnie tak!

Wygłada na to że w dniach 15-17 lipca będzie organizowana pierwsza, mniejsza impreza mająca na celu wypuszczenie stuffu z invitation na właściwe już party, które odbędzie się w dniach 11-13 listopada 2016 roku w Gdańsku. Zegar już odmierza czas na stronie:

www.sillyventure.eu



Outline po raz 13-ty zaprasza atarowców na party

Jak co roku Havoc zaprasza na swoje party do Holandii. Spotkanie jest atarowskie, ale inne platformy są mile widziane. Więcej informacji pod adresem:

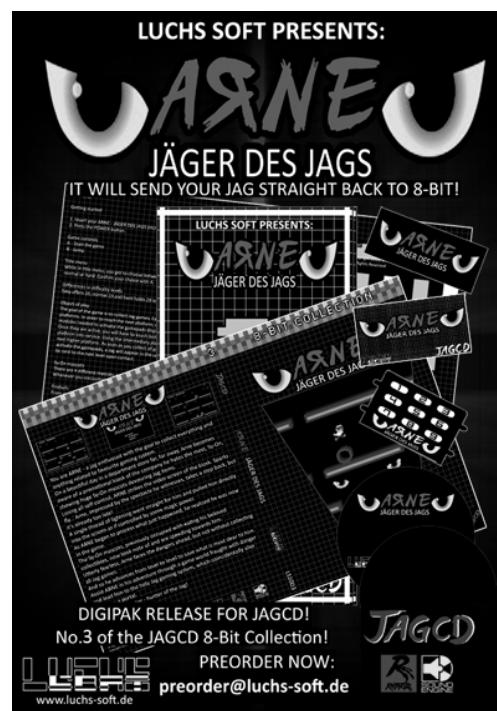
outlinedemoparty.nl

Nowości na Atari Jaguara z Luchs-Soft!

W zeszłym roku ludzie z Luchs Soft dali nam aż trzy tytuły na Jaguara: ABSORBIERER, AUSWEICHMANÖVER oraz ARNE - Jäger des Jags. Są do kupienia na stronie:

www.luchs-soft.de

Wszystkie są bardzo dobrze wydane na płycie CD. Gry są prostymi łami-główkami i zręcznościówkami. W przyszłym numerze Retrokompa znajdą się ich opisy. Mamy już 2016 rok, a chłopaki zapowiadają kontynuację przygód Arniego oraz port jednej z gier, która wyszła już na konsolę Vectrex. Jaki to tytuł na razie nie wiadomo.

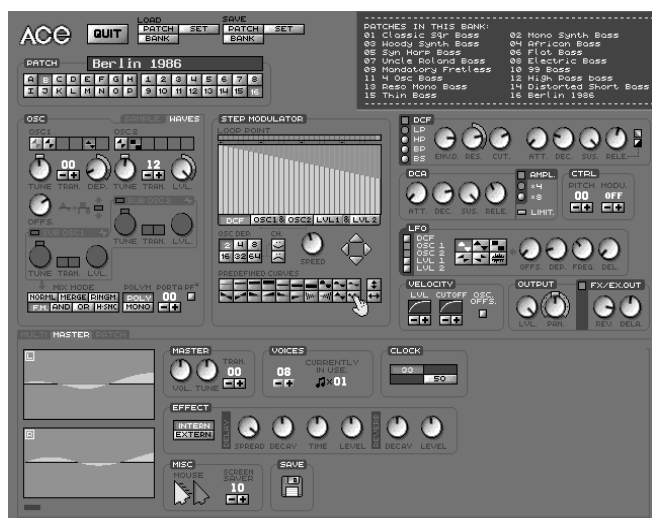
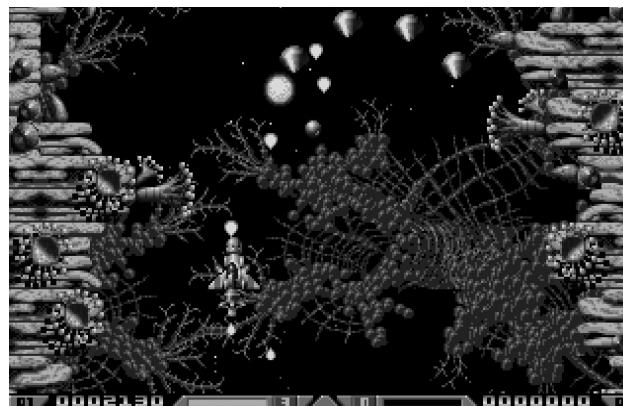




Xenon 2 - wkrótce wersja specjalna na Atari Jaguara

Specjalna, ulepszona wersja Xenona 2 jest już na ukończeniu! Jest to port wersji ST z poprawioną płynnością oraz częściowo zmienioną ścieżką dźwiękową, autorstwa znanego muzyka 505. Gra dostępna będzie na pewno na cartridge, wydana podobnie jak ostatnio opisywany przez nas JHL 15, czyli dokładnie tak, jak oryginalne cartridge atarowskie.

Więcej informacji, a być może opis w następnym numerze naszego pisma.



Ace Tracker 2.0 dla Atari Falcon 030 - nowa wersja

Świetny program dla muzyków doczekał się aktualizacji. Nowa wersja, jak i opisane zmiany tego soundtrackera, który zbudowany jest na silniku ACE, znajdziemy na stronie:

nb.atari.org

Karta turbo do Falcona - CT63e - prace trwają!

Nadal trwają prace nad ulepszoną wersją znanej i pożądanej dopalki do Falcona - CT63e. Karta jest całkowicie przeprojektowana i jak na razie wiadomo nam o dwóch jej prototypach. Są jeszcze problemy z ergonomią - ma być mniej problematycznie instalowana w oryginalnej obudowie komputera. Autorem jest oczywiście Rodolphe Czuba który udostępnił naszemu koledze Willy'emu projekty i pozwolił na produkcję pewnej partii tych urządzeń. Więcej informacji w następnym numerze Retrokompa.

CosmoEx podłączymy także do Falcona i TT!

Rewolucyjne, multifunkcyjne urządzenie służące m.in. jako dysk twardy i karta sieciowa teraz doczekało się wsparcia SCSI! Możemy je więc podłączyć - oprócz komputerów serii Atari ST/STe - także do Falcona oraz TT-ki! zamówienia składać można na stronie:

joo.kie.sk



Maggie 25 - magazyn dyskowy dla Atari ST/STe



Nowy numer tego magazynu dyskowego ukazał się pod koniec 2015 roku, a w nim poczytać można relacje z party takich jak STNICCC 2015 oraz SillyVenture 2k14. Oczywiście są też recenzje gier, programów i nowinki ze świata sceny atarowskiej. Magazyn dostępny jest także w wersji HTML - znajdziecie go pod adresem:

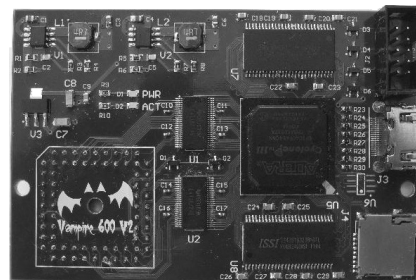
www.dhs.nu



Karta Vampire 600 V2 jest już dostępna!

Pierwsze egzemplarze karty turbo do Amigi 600 z procesorem w technologii FPGA docierają do kupujących. Produkt daje solidne przyspieszenie, osiągające praktycznie ok. 80-100 MIPSów, a więc 2-3 razy więcej niż karty z Motorolą 68060 taktowaną 50 MHz. To nie koniec! Wampir posiada złącze HDMI i oprogramowanie pozwalające używać nowych trybów graficznych z rozdzielczościami nawet FullHD! W przyszłości przewidywane jest dalsze rozszerzanie funkcji firmware i to bez potrzeby zmiany sprzętu. Więcej przeczytacie na stronie 57 oraz w Internecie:

www.majsta.com



Google Drive Handler

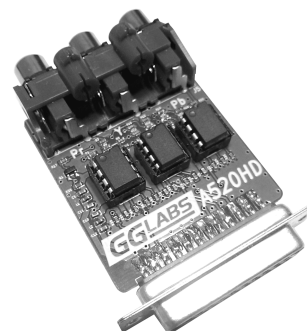
Pojawił się program umożliwiający obsługę usługi Google Drive, czyli popularnego "Dysku Google". Wymagania nie są bardzo duże, potrzebujemy tylko procesora 68020 i kilka megabajtów pamięci Fast. Program korzysta z mechanizmów systemowych, a po uruchomieniu na Workbenchu pojawia się nowe urządzenie, z którego można korzystać tak samo jak ze zwykłego dysku, oczywiście z pewnymi ograniczeniami. Program można znaleźć na Aminiecie, pod adresem:

aminet.net/package/comm/tcp/GoogleDrive_handler_68k

A520HD

Firma GGLabs opracowała zamiennik dla oryginalnego modulatora telewizyjnego Amigi 500. Nowa wersja konwertuje sygnał RGB na wysokiej jakości obraz zgodny ze standardem HDTV (Component Video). Urządzenie kosztuje 70 dolarów, a zamówić można je na poniższej stronie:

gglabs.us



VIC20 - The Penultimate Cartridge



Dla Commodore VIC-20 powstaje nowy cartridge, który będzie umożliwił rozszerzenie pamięci do 32 kilobajtów, a także wczytywanie obrazów ROM. Wszystko będzie można elegancko konfigurować za pomocą przełączników, bez konieczności wyjmowania z komputera.

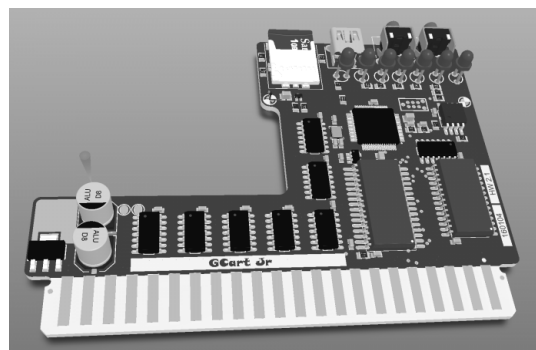
O projekcie będziemy z pewnością pisać więcej, można go już zamawiać na stronie:

blog.tynemouthsoftware.co.uk

GCart Jr.

Tak właśnie nazywa się kolejny cartridge dla Commodore VIC-20. Oferuje 16-megabajtową pamięć wewnętrzną Flash, gniazdo dla karty pamięci micro-SD (do 32 GB) oraz złącze USB w celu połączenia z oprogramowaniem działającym po stronie komputera PC (obsługiwany ma być Windows i Linuks). Za pomocą GCart będzie można więc łatwo wczytywać programy, ma mieć też wygodne menu oraz przycisk Reset. Przewidywana cena to 60 dolarów. Więcej na stronie:

lovskog.net/projects/index.php/category/projects/vic20/gcart_jr/



Przygotował: Piotr "Piter" Krużycki



Wybór dla hobbysty

Paweł "Juen" Nowak
oraz
Krzysztof "Koyot 1222"
Matys
- autorzy gry
"Tanks Furry"

Pytanie na rozgrzewkę: Wasze pierwsze komputery i czy były to przypadkiem Amigi?

Juen: Moim pierwszym komputerem było Atari 65XE, które otrzymałem od wujka zajmującego się sprzedażą sprzętu RTV bodajże w 1990 roku. Największą frajdę sprawiała mi zabawa w Basicu i "udawanie", że napisałem grę. Dwa lata później byłem standardowym (czyli szczęśliwym) posiadaczem Amigi 500.

Koyot 1222: Moim pierwszym "komputerem" była CD32. Ktoś kiedyś nam "doradził", że to taka tańsza, mniejsza i nowsza amiga 1200. Po kilku latach okazało się niestety, że nie da się jej połączyć z drukarką którą nabyliśmy. Za namową brata zakupiliśmy używaną Amigę 1200 w obudowie Tower z kartą turbo, która służy nam wiernie wiele lat.

Jakich Amig używacie teraz? Czy są to systemy bazujące na starym dobrym AmigaOS 3.x czy może jednak coś nowego, typu AmigaOS 4?

Juen: Swoich pierwszych Amig już nie mam, ale parę lat temu nabyłem gołą A1200, potem ubrałem ją w ACA1221. I do teraz przesuwam ikonki na AmigaOS 3.1.

Koyot 1222: Cały czas używam CD32. Teraz, z powodu pracy nad Tanks Furry,

częściej włączam 1200. Ciągłe używam WB 3.1 z ulubioną nakładką Magellan 2. Jak to mówią "szewc bez butów chodzi", system lekki bez nakładek i "upiększaczy"... Nawet tapety nie mam. Sprzęt, który powoli zaczął się kurzyć, nagle odżył.

Dzięki koledze z forum PPA dostałem 2 lata temu w "darze dla developera" sloty Zorro II i kartę graficzną Spectrum - spora różnica w porównaniu z ekranem telewizora. Nie zdążyłem jeszcze zainteresować się nowszymi systemami post-amigowymi, nie zadziałają na moim sprzęcie, chyba zostałem z tyłu. Ostatnio interesowałem się trochę AROSem 68k.

Skąd pomysł na grę? Jak duży zespół wchodzi w skład PROJECT R3D?

Juen: Nie jest to nasza (ani moja) pierwsza zaczęta gra, ale za to pierwsza, która będzie skończona. Uznaliśmy, że zawiesimy chwilowo pracę nad większym projektem i zrobimy grę, którą da się szybko skończyć. Nie ma tu rozlazłej fabuły czy tony dialogów - wy-

starczy zrobić tę przeklętą grafikę, piekielny kod i natrzaskać misji! Cud, nie przepis! Padło na Tanki, które chodziły (jeździły?) po głowie od dawna Koyotowi, co ciekawe - okazało się, że mi również, więc decyzja zapadła szybko. Tym bardziej, że pasowało Wam coś pokazać na RetroKompie.

Koyot 1222: Zafascynowany Advance Wars na konsolę Gameboy dawno zacząłem rysować sobie własne sprajty do gry o czołgach. Jak wiele innych projektów, trafiły "do szuflady". W międzyczasie robiliśmy R3D i zaświtał nam pomysł, aby zrobić coś mniejszego i zdążyć na konkurs retro gier na RetroKomp 2015. W międzyczasie odezwał się do mojego ami-kolegi z bloku obok, który teraz wymienił zajmowanie się Amigą na jazdę tramwajem, żeby poratował nas dźwiękami. Po kilku tygodniach doszliśmy do wniosku, że "takich dwóch jak nas trzech, to ani jeden" i zabraliśmy się do pracy.

Jak już kurz opadł, zamieściliśmy filmik z naszego demo retrokompowych czołgów. Zaczepił nas znany w środowisku JMD, który wspomógł nas świetną mu-



zyką chiptune. Podbudowani bardzo pozytywnym przyjęciem gry, stwierdziliśmy, że skoro mamy już tak wiele, trzeba projekt skończyć. W międzyczasie kilka osób wypatrzyło dziury w naszym tłumaczeniu, potem pojawił się Aszu, który bardzo rzetelnie podszedł do korekty tekstów.

W krótkim czasie lista twórców rozrosła się do 5 osób, ale to nie wszystko. Cały czas, w którym my tworzyliśmy grę, Zeter wraz z Pixelnation użytkował na produkcję nowoczesnego trailera, który w ciągu kilku dni uzyskał około 5 tys. odsłon na kanale YouTube. Mamy dość sporą grupę ludzi skupionych wokół forum PPA, którzy nam bezinteresownie pomagają. Raz są to bardzo dobre uwagi odnośnie grafiki, innym razem kodu, korekty napisów itp. Doliczyc też trzeba kolejną grupę osób, które zainwestowały czas, żeby w różnych miejscach Internetu powiesić informację o naszej grze. Zważywszy, że jest to projekt od początku non profit, pomoc i zainteresowanie naszą pracą mocno przeszło oczekiwania. Serdecznie wszystkim dziękujemy!

Ile czasu zajęła wam praca nad grą? Czy gra powstawała w pełni na Amigach, czy musieliście posiłkować się komputerami, nazwijmy to, zastępczymi?

Juen: 50% kodu powstało w mniej niż dwa tygodnie, później na ile tylko już czas pozwalał. Traktując to zawodowo myślę, że nie całość trwała nie dłużej niż dwa miesiące - wliczając w to testy i poprawki. We wklepywaniu kodu wygrał Notepad++ na PC, przy którym mam podpięte 4 monitory specjalnie do celów programistycznych. Resetowanie środowiska na Amidze co moment, żeby edytować jedną linijkę i tak w kółko, byłoby sporym utrudnieniem, tak więc na jednym z monitorów wisiło WinUAE (020 + Fast), gdzie po każdym resecie odpalał się Asm-One z autostartem ładowania głównego pliku, asemblacji i uruchomienia. Co jakiś czas plik wykonywalny leciał na leżącą obok prawdziwą Amigę w celach testowania.

Koyot 1222: Tanks Furry tworzony był w wolnym czasie, przez kilka miesięcy. Pierwsze propozycje grafiki zacząłem wrzucać na forum ponad pół roku temu.

Myślałem o tym aby grafika powstawała częściowo w Personal Paint, ale niestety awaria karty graficznej przekreśliła ten plan. Grafika w całości powstała na pececie przy użyciu, GIMPa. Photoshop służył mi często jako program do podglądu docelowych animacji. Pomimo wieloletniego doświadczenia przy

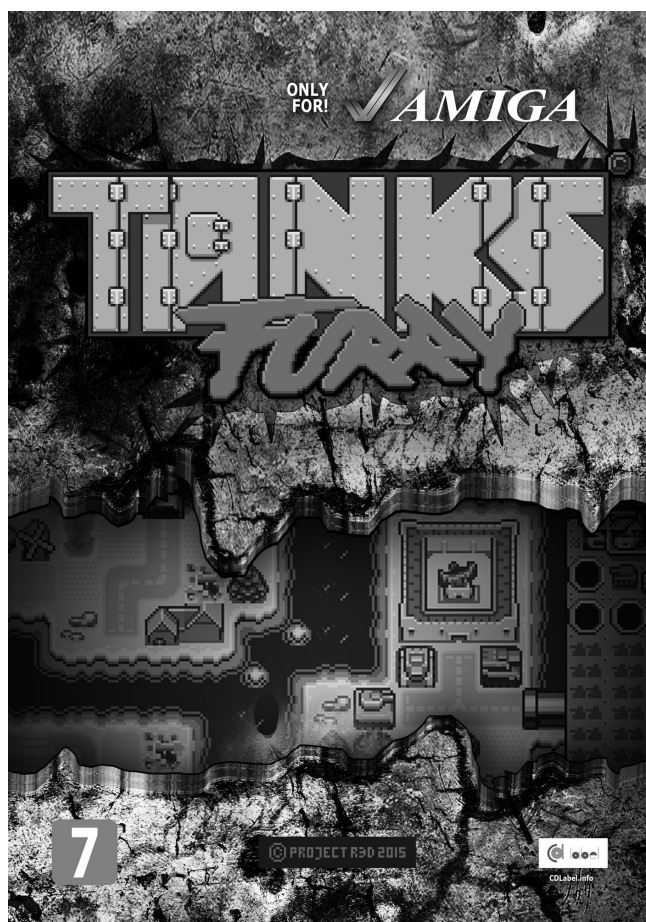
obróbce do DTP nie potrafię w nim robić pikseli, ma też poważne ograniczenie braku wielowarstwowości przy projektach indeksowanych. Przygotowując grafikę do Fruity przyzwyczaiłem się też do konfigurowalnej siatki GIMPa, która wygodnie rozdziela mi elementy projektów.

Ostatecznie nie obyło się jednak bez amigowego Personal Painta. Okazał się niezawodnym narzędziem do konwersji przygotowanej grafiki, podmiany palety itd. Pliki podróżują między moim PC z okienkami a kartą CF pod WinUAE do Amigi - do stacji dysków wymiennych, tam i z powrotem. Przy tworzeniu plansz Juen napisał 3 rodzaje edytorów plansz online. Narzędzia te skróciły pracę nad misjami o kilkadziesiąt procent. Wykorzystał system tworzony przy R3D, ale dodane zostały inne potrzebne elementy. Taki system bardzo mi odpowiada, jest w tym pewna dawka nowoczesności, przepleciona starodawnością. System, który niektórych przyprawiłby o siwe włosy, dla mnie jest bardzo wygodny. Coś, co bardzo dobrze działa od lat na scenie C64, gdzie za pomocą coraz nowocześniejszych narzędzi powstają świeże dzieła, nie tylko w dziedzinie dem.

"Tanks Furry" właściwie momentalnie zostało ciepło przyjęte w ami-środowisku. Wasza produkcja prezentuje się świetnie, a rynek jest wygłodniały. Czy mimo to jesteście zaskoczeni takim obrotem sprawy?

Juen: Bardzo jesteśmy zaskoczeni i naprawdę mamy nadzieję, że zacznie powstawać więcej produkcji, nie tylko na RetroKompa - aczkolwiek jest to dobre miejsce do pokazania nad czym się aktualnie pracuje.

Koyot 1222: "Kariera" Tanks Furry przypomina trochę śniegową kulę. Musimy pamiętać, że jest to tylko gra. Od początku staraliśmy się informować tylko o rzeczach zrobionych i działają-



cych. Postanowiliśmy zacząć od złamania pewnej ami-tradycji planowania rzeczy i niedotrzymywania obietnic. Trudno teraz mówić jeszcze o jakimś sukcesie. Wydaje się mi, że ludzie zawsze czekali na nowe gry na Amigę, a ich po prostu długo nie było.

Bardzo cieszę się, że gra jest wizualnie atrakcyjna, włożyłem w to wiele pracy. Jest to rzecz, która rzuca się na pierwszy rzut oka. Bardzo cieszy mnie porównywanie grafiki stworzonej od podstaw dla Tanks Furry z produkcją wybitnego Advance Wars. Pod tym "nakryciem" skrywa się turbo doładowany assemblerowy silnik, który porusza minimum 256 animowanymi elementami na ekranie.

Jeśli doliczymy do tego jeżdżące animowane czołgi, wybuchy, wystrzały to okazuje się, że w sekundę mamy ponad 1000 (tak, aż tysiąc!) cykli na sekundę na ekranie gołej Amigi z 1 MB pamięci. 42 misje solo, 15 misji multi, intro, 4 zakończenia gry, muzyka i efekty zmieściły się na 1 rzadkiej dyskietce. Uważam to za ogromny sukces, nie ujmując nic muzyce, która powstała specjalnie dla naszej produkcji.

Czy Tanks Furry będzie działać natywnie pod nowszymi systemami jak MorphOS czy AROS?



Juen: Gra jest napisana pod hardware amigowy, tak więc można powiedzieć, że zadziała tylko tam, gdzie inne typowe stare produkcje.

Koyot 1222: Nie chciałbym wchodzić w kompetencje Juena, ale wydaje mi się, że emulacja na tych systemach rozwiąże sprawę wszelkiej niezgodności. Miło będzie nam jeśli gra dotrze na te systemy i zostanie także ciepło przyjęta, a nasza praca doceniona.

W związku z tym czy Wasze następne gry będą wychodzić poza sferę Amigi klasycznej? W ogóle jakie gry planujecie? Żartobliwie powiem, że teraz nie macie już wyjścia, musicie dopieścić graczy kolejnymi pozycjami.

Koyot 1222: Przykład partnerstwa IndieGo, pokazał nam nieoczekiwanie, że nasza gra może nieoczekiwanie wskoczyć na pole zupełnie przez nas niezaplanowane. Chyba będzie to pierwszy przypadek od 20 lat, kiedy gra debiutu-

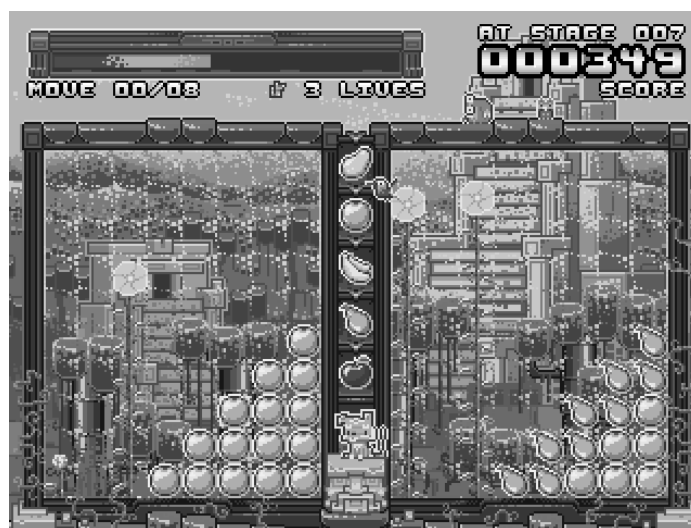
Juen: Emulacja pozwala na wszystko więc sprawa otwarta.

Koyot 1222: Wydanie pudełkowej Tanks Furry to trochę przypadek. Po ogłoszeniu zbliżającego się końca pracy nad grą i udostępnienia jej, według wcześniejszych zapowiedzi - za darmo, wiele osób zaoferowało nam pomoc w postaci dotacji. Nie bardzo wiedzieliśmy w pierwszej chwili co z tym zrobić. Jak przyjmować pieniądze i na co je przeznaczyć. Po rozważeniu ewentualnego wydania pudełkowego, okazało się, że zgłosiły się do nas osoby chcące pomóc w dystrybucji i sprzedaży.

Od początku tworzenia R3D i Tanks Furry, nie mieliśmy pomysłu na sprzedaż naszych gier i zarabianiu na nich a co dopiero konwertowaniu naszych produkcji i opatowywania rynku, na którym profesjonaliści bardzo dobrze sobie radzą.

Juen: Chyba na razie nie planujemy szybkiego rozrostu grupy, chociaż możliwe że wkrótce niektóre projekty będą tego wręcz wymagały.

Koyot 1222: Przy grafice do Tanks Furry pracowałem praktycznie sam, podobnie jak wcześniej przy Fruity. Podobnie Juen, który w każdym wolnym czasie kleił kod. Nawet przy takim niby niewielkim projekcie szybko okazało się, że do pewnych rzeczy nie mamy talentu ani cierpliwości. Dzięki szczęśliwym zbiegom okoliczności na naszej drodze pojawiły się kolejne osoby, które pomagają nam na każdym etapie tworzenia ratując nas dźwiękami lub muzyką (JMD, autor muzyki do Tanks Furry jest Włochem mieszkającym w USA, a Marcin od SFXów nadal mieszka w bloku obok mnie). Mamy też



Czy uważacie, że środowisko Amigi jest gotowe na powrót komercyjnych gier?

jąca na Amidze pojawi się na innym systemie. Jest to trochę przypadek, znalezienia się w odpowiednim miejscu o odpowiednim czasie.

Chcielibyśmy dalej tworzyć gry, szczególnie na retro platformę 68k. Czy będą to pozycje równie popularne jak Tanks Furry to jeszcze się okaże. Mamy ten komfort, że tworzymy z pasji, a nie na zamówienie. Plany na rok 2016 to głównie skupienie się wokół projektu i dokończenie R3D. Mamy bardzo pozytywne doświadczenia z pracy przy Tanks Furry, a więc jest dla kogo tworzyć takie produkcje.

Myśleliście nad wydaniem Tanks Furry (i przyszłych projektów) na inne platformy, wykorzystując np. Steam?

Juen: W niewielkim nakładzie - zawsze. Sam bym kupował, najchętniej w wersji fizycznej, byleby koszt przesyłki mnie nie zjadł.

Koyot 1222: Nasza gra powstała z chęci stworzenia czegoś nowego na Amigę. Zaplanowana sprzedaż pudełkowa ukazała spore zainteresowanie tym tematem. Mamy świadomość, że nie wskresimy ani samej Amigi, ani chociaż części jej legendy. Mamy jednak nadzieję, że złoty ami-sympatyków zaczęną być bardziej barwne, chociażby dzięki naszej produkcji "multiplayer". Niczym sympatycy starych klasycznych samochodów, możemy zaprezentować, że da się stworzyć coś na nowo.

Czy potrzebujecie zdolnych ludzi, którzy chcieliby dołączyć do tak ciekawej inicjatywy jak PROJECT R3D?

w gronie kilka osób które chętnie rozsyłają informacje o grze, nie zapominając kolegi Zetera z Pixelnation który zrobił nam świetny trailer! Wszystko to komponuje się w wielki zespół, który tak naprawdę nie jest zespołem, a luźną rodziną sympatyków retro komputerów. Pomocne ręce są zawsze dobrze widziane!

Co myślicie o środowisku Amigi? Podział na sprzęt retro oraz na aż trzy systemy nowej generacji nie ułatwia tworzenie oprogramowania.

Juen: Przynajmniej jest wybór, nikt nie będzie też oczekiwał od programisty robienia czegoś od razu "na trzy". Zdolnych programistów i koderów było zawsze wielu, jednym bardziej pasuje klasyczna Amiga (tak jak mnie), a inni zajmą się nowymi OSami. Ostatecznie są to systemy wyłącznie dla hobbystów, więc trzeba mieć nadzieję, że wśród nich są także osoby chcące coś stworzyć, a nie tylko używać.

Koyot 1222: Aktualną sytuację porównałbym do zjazdu Volkswagena Beetle. Niektórzy wolą klasykę w czystym spartańskim wydaniu, inni turbo doładowane jednostki o zupełnie nowych możliwościach, znowu inni mający dość całego tego zgiełku przesiadają się na "nju" Beetle. To wydaje się zdrowe poście.

Cieszę mnie faktem, że skostniałe przez wiele lat środowisko skupione wokół amigi - tej legendarnej retro i nowej, zaczyna tworzyć coś swojego. Pojawiają się coraz to nowsze peryferia, co roku mniejsze i większe gry, za sprawą konwersji, ale też napisane od podstaw (np. tegoroczne świetne Zerosphere). Może nie jest to jeszcze taki ruch jak na C64, ale jestem dobrej myśli. Mam nadzieję, że słońce dopiero wschodzi.

Czy nadchodząca "rewolucja" w postaci kart FPGA, emulujących procesory serii 68k pomoże w odrodzeniu rynku oprogramowania dla Amigi?

Juen: To pokażą dopiero najbliższe lata, czy ich cena spadnie i ilość dostępnych kart będzie wystarczająca. Jeśli użytkowników przykładowego Vampira 2 będzie minimum parę tysięcy to jasne, ale głównie będą to porty znanych programów.

Koyot 1222: Ogólnie jest to wspaniała wiadomość dla użytkowników wysłużonych już kart turbo i innych klasycznych peryferiów. Ceny sprzętu z lat '90-tych dosyć mocno poszybowały w górę, ale chyba nastąpił martwy punkt. Pojawienie się nowych kart opartych na FPGA w rozsądnej cenie być może ożywi ruch i pozwoli na nasycenie spragnionego

rynku. Amiga klasyczna to wiele ciekawego oprogramowania, jest od czego zacząć.

Końcówka lat 90-tych, mimo wszystko dla poziomu gier dostępnych dla systemu AmigaOS była bardzo dobra. Czy widzicie szansę na tworzenie teraz (lub w niedalekiej przyszłości) takich produkcji jak np. Exodus, Foundation czy Nemac IV?

Juen: Oczywiście! Jeśli tylko wszystko pójdzie w dobrym kierunku i będzie zainteresowanie grami na mocniejsze Amigi. Ja osobiście z chęcią zrobiłbym przynajmniej jedną grę z każdego gatunku.

Koyot 1222: Jesteśmy bardzo małym zespołem, jeszcze zbyt małym na robienie tak rozbudowanego projektu jak np. Exodus. Jeszcze miesiąc temu nie myśleliśmy nawet o Tanks Furry 2. Mierzmy siły na zamiary, zobaczymy co czas przyniesie.

Czego wam najbardziej brakuje w Amidze? Nie ważne czy mówimy o klasyku czy o sprzęcie nowej generacji.

Juen: Szybszego blittera/pamięci CHIP oraz 8 kanałów audio. To by było...

Koyot 1222: Od kilku lat obserwuję scenę C64, może nawet lekką nutką zawiści. Wielobarwna scena C64 na której cyklicznie pojawiają się ciekawe, niekoniecznie "super duper uber" nowoczesne produkcje. Dema, gry, muzyka, grafika, twórczość która dosyć mocno wyewoluowała przez lata. Ja też szu-



kam sobie jakiegoś miejsca i drogi na scenie retro-ami, wykorzystując, łącząc styl i technologię z poza tego kręgu. Spotykam co jakiś czas podobnych pasjonatów w innych dziedzinach jak programowanie, muzyka itp. Fajnie jak będzie takich ludzi jeszcze więcej, zbierających się w grupy pod podobnym szyldem jak Project R3D. Może niektórzy zbyt ambitnie podchodzą do swoich prac, a przecież chodzi głównie o samorealizację i dobrą zabawę.

I na sam koniec. Retro powraca z ogromną siłą w każdym z obszarów życia. Jak widać Amiga może na tym skorzystać. Jakie jest wasze zdanie?

Juen: Myślę, że zarówno Amiga jak i każdy inny komputer czy konsola już na tym zyskują. Mam tylko nadzieję, że będzie to trwało w najlepsze.

Koyot 1222: Mam nadzieję, że zainteresowanie retro sprzętem nie przeminie tak szybko jak moda np. na szerokie spodnie. Mam nadzieję, że ami-złoty i ogólnie fanów retro sprzętu zaczną przypominać złoty wielbicieli klasycznych samochodów. Gdzie zarówno ten, który lubi stare dyskiety i poślizgnięte pudełka i ten, który podłącza do swojej Amigi nowoczesną kartę mega-turbo, znajdują swoje miejsce i cieszą się równą akceptacją.

Kiedy byłem raz w muzeum w Luton w Wielkiej Brytanii, widziałem za szybą zafoliowane stare komputery i konsole. Pamiętam, że miałem mieszane uczucia, trochę jak głodne dziecko, które stoi przed wystawą z cukierkami. Cudowne, że fascynacja retro sprzętem rozwija się wokół wyciągania tych komputerów, odrestaurowania, używania i dobrej zabawy.

**Rozmawiał:
Krzysztof "Radzik" Radzikowski**





Tryby graficzne TIMEXA

Timex Sinclair 1000 to pierwszy komputer będący owocem współpracy joint-venture między amerykańską firmą Timex oraz Sinclair Research. Pojawił się w 1982 roku i był kłosem ZX81 - popularnego na Wyspach następcy ZX80. Na kolejne owoce współpracy nie trzeba było długo czekać i tak na rynku pojawiają się: wyprodukowany pod koniec roku 1983 - Timex Sinclair 2068, a następnie produkowane już w Europie od 1984 roku - Timex Computer 2068 i Timex Computer 2048. Komputer ten jest w 99% kompatybilny z oryginalnym ZX Spectrum, a do tego oferuje coś ekstra – dodatkowe tryby graficzne.

Możliwości ma Spectruma...

Zastosowany układ PAL2068 (odpowiednik ULA z ZX Spectrum) potrafi wyświetlić dwukolorową grafikę w rozdzielczości 512x192 pikseli, 15 (a niektórzy twierdzą że i 16) kolorową grafikę w rozdzielczości 256x192 z większą gęstością kolorów, tj. kafelki 8x1 zamiast 8x8. Dodatkowo udostępnia programiście drugi obszar pamięci video, który znamy z ZX Spectrum 128 (wyprodukowany dopiero w 1986r).

W niniejszym artykule przybliżę sposób organizowania i funkcjonowania dwóch z trzech dodatkowych trybów graficznych. Ze względu na pewien stopień zawartości omawianej tematyki, początkujący użytkownicy mogą mieć nieco trudności ze zrozumieniem niektórych kwestii, dlatego byłoby dobrze, żeby przed przeczytaniem posiadali już pewną wiedzę na temat organizacji pamięci graficznej w ZX Spectrum.

Emulatory Timexa i edytor grafiki

Do testów dodatkowych trybów graficznych najlepiej używać prawdziwego sprzętu. Jeżeli nie mamy takiej możliwości, musimy do tego celu użyć emulatorów. Istnieje kilka emulatorów ZX Spectrum, które emulują również Timexa. Są to między innymi: Fuse, dostępny pod adresem:

fuse-emulator.sourceforge.net

moim zdaniem najlepszy, SpecEmu:

homepage.ntl-world.com/mark.woodmass/specemu.html

oraz Eighty One:

www.chuntey.com

W każdym z nich należy wybrać Timexa spośród emulowanych modeli komputerów. Jeżeli dodatkowo interesuje nas tworzenie grafiki, to warto zainteresować się ZX Paintbrush:

<http://www.zx-modules.de/zxpaintbrush/zxpaintbrush.html>

czyli programem graficznym dla systemu Windows oferującym tworzenie, konwersję grafiki w standardowym trybie ZX Spectrum, jak również obsługę dodatkowego trybu graficznego, zwanego multikolorem.

Port 255 (\$FF)

Do ustawiania trybów graficznych służy port 255. Wykorzystywane są w nim wszystkie bity, lecz nas interesować będzie jedynie sześć pierwszych. Ich rozpiska widoczna jest na rys. 1.

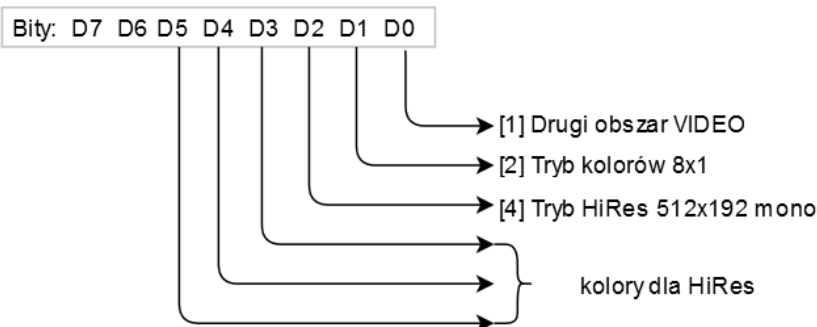
Każdy z bitów, zaczynając od najmłodszego, zostanie omówiony przy opisywaniu poszczególnych trybów graficznych.

Zaczynamy!

Dodatkowy obszar pamięci VIDEO

Bit zerowy ustawia obszar pamięci, z której ma być wyświetlany obraz. Jeżeli bit ma wartość 0 to obraz jest wyświetlany z adresu 16384 (\$4000), jeżeli ma wartość 1 to z obszaru pamięci rozpoczynającego się od adresu 24576 (\$6000). Istotną informacją ułatwiającą oprogramowanie drugiej pamięci ekranu oraz pozostałych trybów

Port 255 (\$FF)



Rys. 1. Tryby graficzne portu 255.

graficznych jest fakt, że rozpoczyna się dokładnie 8192 (\$2000) bajty po pierwszej. Organizacja pamięci jest w obu przypadkach identyczna. Pierwsze 6144 bajty to dane o pikselach, kolejne 768 bajtów to informacja o mapie kolorów (atrybutach).

Aby włączyć wyświetlanie z dodatkowego obszaru VRAM, należy wydać polecenie (tu w BASIC):

```
OUT 255,1
```

Powrót do podstawowej pamięci następuje po:

```
OUT 255,0
```

Aby przekonać się, że sposób działania drugiego ekranu jest identyczny, jak z pierwszym, należy napisać prosty program w BASIC (przykładowy listing poniżej).

```
10 CLEAR 24575: BORDER 0
20 OUT 255,1
30 LET src=0: LET
   add=8192
40 FOR d=23295+add TO
   16384+add STEP -1
50 POKE d,PEEK src
60 LET src=src+1
70 NEXT d
80 PAUSE 0: OUT 255,0
```

Co on robi? Kopiuje zawartość ROM do pamięci drugiego obszaru VIDEO. Robi to od końca pamięci ekranu, czyli najpierw zapełnia obszar atrybutów, a następnie stawia piksele. Zaczynamy od atrybutów, aby szybciej uwidocznić działanie programu, gdyż po starcie komputera w pamięci kolorów zapisana jest wartość 0 - co przekłada się na czarny kolor pikseli i tła. Po skończeniu działania, czeka na naciśnięcie dowolnego klawisza, a następnie wraca do podstawowej pamięci VRAM.

Jeżeli zmodyfikujemy linię 30 poprzez zmianę wartości zmiennej "add" na 0 oraz komentując (dodając instrukcję REM na początku linii) lub usuwając linię 20, możemy obejrzeć ten sam efekt dla podstawowej pamięci ekranu w ZX Spectrum.

Bardziej praktyczne zastosowanie dodatkowego VRAM prezentuje kolejny program. Ładuje on jeden obrazek do podstawowego obszaru pamięci wideo (adres 16384), a następnie do dodatkowego (adres 24576). Po załadowaniu



Rys. 2. Fragmenty dwóch obrazków po załadowaniu do różnych obszarów pamięci.

danych następuje, w nieskończonej pętli, przełączanie wyświetlania obrazu z obszarów VIDEO RAM: podstawowego i dodatkowego, czego widocznym efektem są fragmenty dwóch różnych obrazków. Widać to na rys. 2.

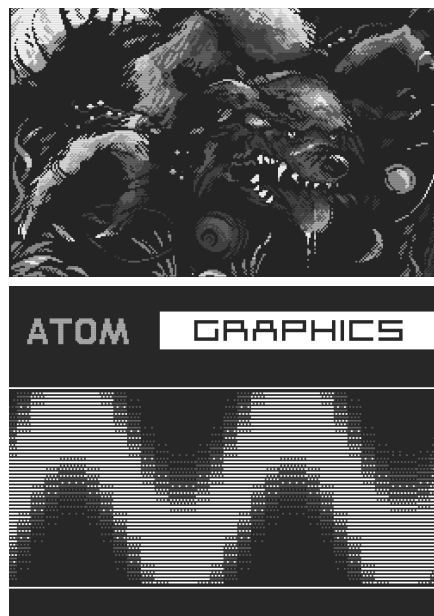
```
5 CLEAR 24575: BORDER 0
10 LOAD ""CODE 24576
20 LOAD ""CODE 16384
30 OUT 255,0: OUT 255,1:
   GO TO 30
```

To tylko przykład, który ma zainspirować do eksperymentów. Można to samo próbować robić w BASICu, lecz aby wyszło z tego coś efektownego, powinno się napisać program w assemblerze Z80. Dodatkowo należy mieć odpowiednio przygotowane obrazki lub też generować je w czasie rzeczywistym, a w odpowiednim momencie modyfikować bit 0 na porcie 255.

Kolory w obszarze 8x1

Ten tryb często określa się jako multicolor. Jego główną cechą jest to, że definiuje kolory w obszarze 8x1 pikseli zamiast standardowo 8x8. Pełnię jego możliwości pokazuje demo Timmy wydane przez grupę Specy.pl w 2014 roku z okazji 30-lecia wypuszczenia na rynek modelu Timex 2048. Na rys. 3. przedstawiono zrzuty ekranowe z tego demo.

Bardziej szczegółowa mapa kolorów niesie za sobą jednak określone konsekwencje. Standardowo w ZX Spectrum atrybuty (kolory) zajmują 768 bajty (32 bajty w 24 liniach), lecz dla trybu 8x1 to aż 6144 bajty (32x192), czyli tyle samo, co obszar pamięci dedykowany pikselom. Jak łatwo można się domyślić, dane kolorów znajdują się pod adresem 24576, a ich organizacja jest identyczna z organizacją pikseli, a nie jak atrybutów. Dla programisty oznacza to tylko tyle, że adres dla kolorów oblicza dodając 8192 do adresu pikseli.



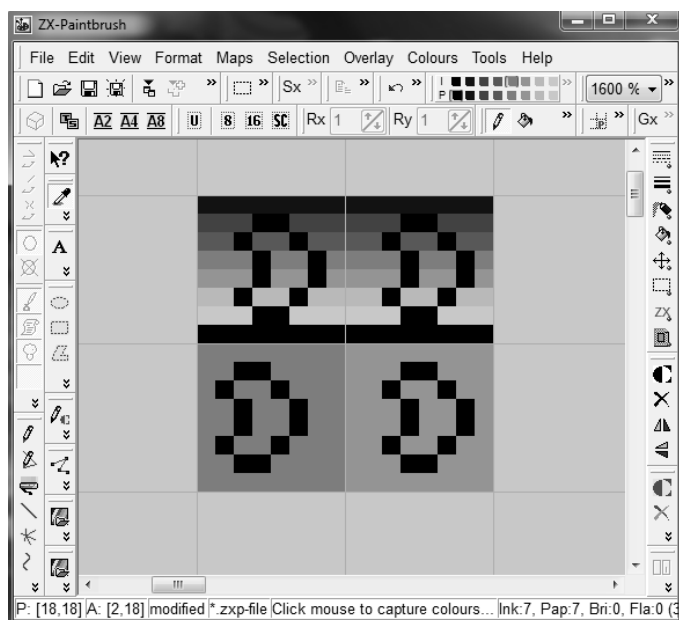
Rys. 3. Demo Timmy / Specy.pl



Należy sobie uświadomić fakt, że teraz do obsłużenia będziemy mieć nie 6144 (6912 z kolorami) bajtów pamięci, lecz aż 12288. Tworzenie zaawansowanych efektów wymaga zmodyfikowania co najmniej dwukrotnie większej ilości pamięci RAM, niż to miało miejsce w standardowym trybie graficznym.

Dla lepszego zobrazowania problemu podpowiem, że procesor Z80 taktowany zegarem 3.5MHz (takim jak w ZX) jest w stanie w ciągu 1/50 sekundy zmodyfikować od 2 do 4 kB danych w zależności od użytej metody przenoszenia danych do VRAM. Mało? Nie jest jednak tak źle, co dobitnie pokazuje wspomniane wcześniej demo.

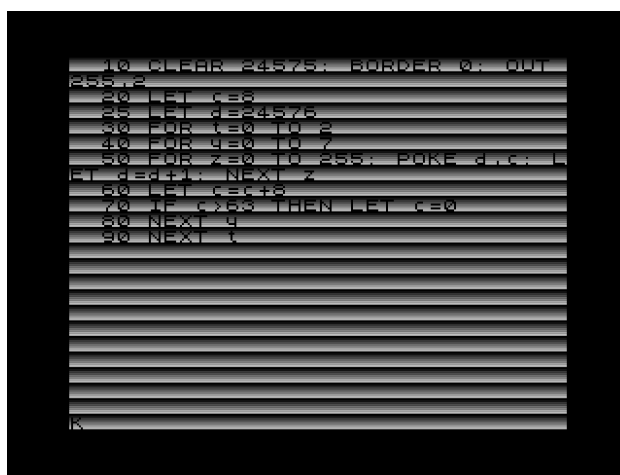
Różnice w grafice prezentuje zrzut ekranu z programu ZX-Paintbrush, gdzie w powiększeniu widać różnicę między 2 kolorami w obszarze 8x8 pikseli (na dole), oraz w 8x1 (na górze).



Rys. 4. Powiększenie obszaru 8x8 pikseli w programie ZX-Paintbrush.

Do włączenia trybu 8x1 służy bit 1 na porcie 255. Bit ten przełącza pobieranie danych o kolorach, a zarazem włącza dodatkowy tryb. Jeżeli jego wartość jest równa 0 to kolory pobierane są z adresu 22528 i są wyświetlane jako 8x8, zaś kiedy wartość bitu równa się 1 to dane czerpane będą z adresu 24576 i wyświetlane jako 8x1.

Obok znajduje się krótki program demonstrujący sposób organizacji pamięci kolorów, a przy okazji generujący kolorowe paski, świetnie widoczne na tle listingu. Ustawiany jest kolor tła (PAPER), zaś kolor pikseli (INK) pozostaje czarny.



Rys. 5. Działanie naszego programu na ekranie komputera.

Włączenie trybu odbywa się poprzez:

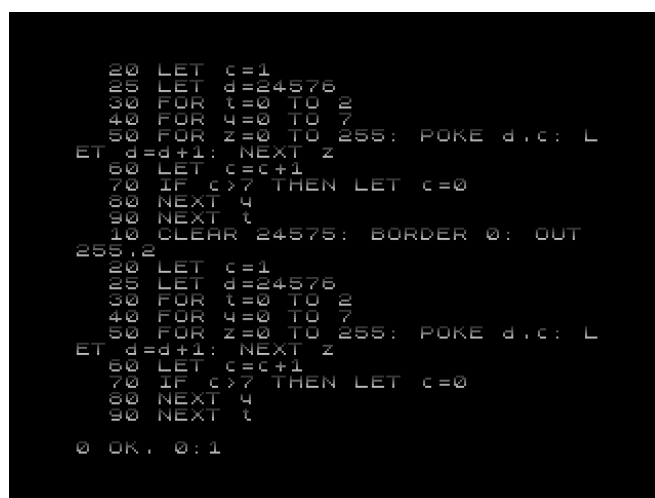
```
OUT 255,2
```

Możemy zmodyfikować program tak, aby działał odwrotnie niż poprzedni, czyli kolor pikseli był zmienny, zaś kolor tła stały. Aby uzyskać taki efekt, należy zmienić następujące linie:

```
20 LET c=1
60 LET c=c+1
70 IF c > 7 THEN
  LET c=0
```

Po uruchomieniu programu otrzymamy efekt jak na rys. 6.

Program może być zmodyfikowany również w ten sposób, że kolory pikseli i tła będą zmienne. To jednak pozostawię jako ćwiczenie dla żądnych eksperymentów Czytelników.



Rys. 6. Program po modyfikacjach.

A co się stanie, jeżeli włączymy również drugą stronę VRAM? Wtedy piksele oraz kolory będą pobierane z dodatkowego obszaru pamięci i na ekranie nie zobaczymy nic sensownego. Jeżeli chcemy się o tym przekonać wystarczy

```
10 CLEAR 24576: BORDER 0:
  OUT 255,2
20 LET c=8
25 LET d=24576
30 FOR t=0 TO 2
40 FOR y=0 TO 7
50 FOR z=0 TO 255: POKE
  d,c: LET d=d+1: NEXT z
60 LET c=c+8
70 IF c > 63 THEN LET c=0
80 NEXT y
90 NEXT t
```



po wykonaniu któregoś z wyżej przedstawionych programów wydać polecenie:

```
OUT 255,3
```

ZX-Paintbrush i format .mtl

Program pozwala na tworzenie i zapisywanie obrazków w trybie Timex Multicolor 8x1. Format ten jest dość prosty, a sposób obsługi w assemblerze przedstawię dalej. Aby zapisać plik w formacie .mtl, należy najpierw utworzyć obrazek w tym trybie. Można to zrobić na dwa sposoby.

Pierwszy polega na utworzeniu nowego obrazka i wybranie "Timex Screen Files (*.mtl)", drugi polega na konwersji plików JPG/PNG/GIF do trybu Timex. Niezależnie od tego, którą opcję wybierzemy, po tym należy przy zapisie wybrać "Timex Screen files (*.mtl)".

Sam plik .mtl ma długość 12288 bajtów. Jego struktura jest następująca:

- pierwsze 6144 bajty to dane o pikselach
- kolejne 6144 bajty to informacja o kolorach.

Z pikselami nie powinno być problemu - ich organizacja jest identyczna jak organizacja pamięci wideo, zatem wystarczy je skopiować bez zmian do pamięci obrazu.

Atrybuty niestety już nie są ułożone zgodnie z organizacją pamięci. Są ułożone liniowo, tak samo jak standardowa pamięć kolorów. Należy zatem przerzucać dane w odpowiedniej kolejności obliczając dla każdego 32 bajtów nowy docelowy adres pamięci. Nie jest to trudne, co demonstruje program w assemblerze mojego autorstwa zamieszczony obok.

Program oczekuje danych o obrazku pod nazwą 'obrazek.mtl'. Kompilacja i wygenerowanie pliku .tap z loaderem basicowym:

```
pasmo --tapbas de-  
codeMTL.asm pro-  
gram.tap
```

W następnej części artykułu, który powinien ukazać się w kolejnym numerze RetroKomp, zajmiemy się trybem Hi-Res.

Krystian "Tygrys" Włosek

```
; Timex Multicolor 256x192 8x1  
; Tygrys / speccy.pl  
;  
    org 32768  
    xor a  
    out (254),a  
  
    ; ustawianie trybu multicolor  
    ld a,2  
    out (255),a  
  
; Pierwsze 6144 bajty to pixele, zgodne  
; z organizacją pamięci ZX Spectrum  
    ld hl,obrazek  
    ld de,16384  
    ld bc,6144  
    ldir  
  
; Kolorы zapisywane są liniowo, a w pamięci  
; muszą być zorganizowane w taki sam sposób  
; jak piksele, zatem przeliczamy adres URAM  
    ld de,16384 + 8192  
    ex af,af'  
    ld a,192  
loop:  
    ex af,af'  
  
    ld bc,32  
  
    push de  
    ldir  
    pop de  
  
    inc d  
    ld a,d  
    and 7  
    jp nz,$+14  
    ld a,e  
    add a,32  
    ld e,a  
    jp c,$+7  
    ld a,d  
    sub 8  
    ld d,a  
  
    ex af,af'  
    dec a  
    jp nz,loop  
    ret  
  
obrazek:  
    incbin "obrazek.mtl"  
end 32768
```



BASIC na ZX Spectrum był dla wielu pierwszym językiem programowania jaki poznali. Po latach wielu chciałoby do niego powrócić i przywołać wspomnienia z młodości, ale na używanie oryginalnego sprzętu nie ma czasu, a emulator nie daje wygody, jakiej się oczekuje od współczesnych narzędzi do programowania. Tutaj z pomocą przychodzi program pod Windows - BASin, którego autorem jest Paul Dunn. Jest on edytorem, kompilatorem, emulatorem ZX Spectrum w jednym. W niniejszym artykule postaram się przedstawić najważniejsze cechy BASin z wielu, jakie powiada ten program, a nie jest ich mało.

Instalacja i uruchomienie

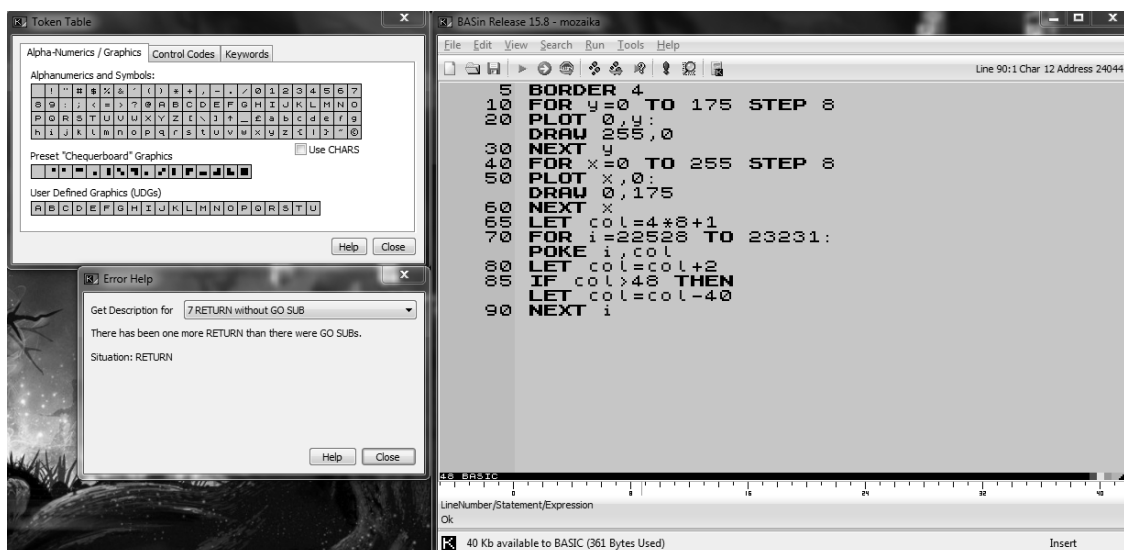
Strona domowa projektu to:

<https://sites.google.com/site/ulaplus/home/zx-spin-and-basin>

zaś pobranie programu nastąpi poprzez kliknięcie na niebieską strzałkę skierowaną w dół na prawo od:

BASin_15_8.zip

Jak informuje autor, projekt nie jest już rozwijany, co jednak nie przeszkadza w jego używaniu. W ściągniętym archiwum znajduje się program oraz jego źródła w Borland Delphi 5.



...a program wpisujemy prawie tak samo, jak na prawdziwym ZX Spectrum.

BASin

wygodny edytor i emulator ZX BASIC

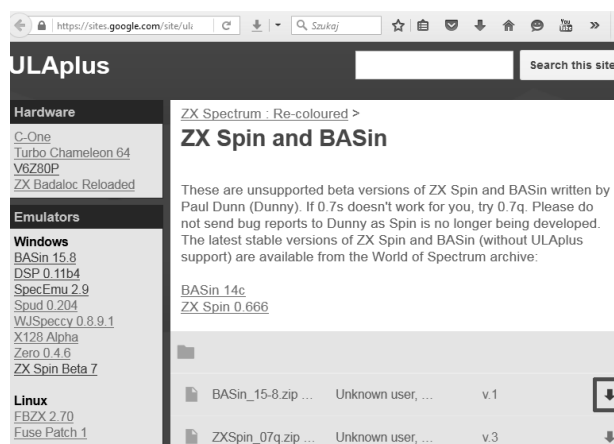
Sama instalacja jest prosta i polega na rozpakowaniu zawartości archiwum do dowolnego miejsca na dysku. Po tym, należy kliknąć w ikonkę programu aby go uruchomić i to wystarczy do rozpoczęcia programowania w ZX BASIC.

Użytkowanie

BASin nie jest prostym edytorem tekstu wraz z zintegrowanym emulatorem. To kombajn oferujący wiele zaawansowanych funkcji, jakie posiadają współczesne IDE (Integrated Development Environment).

Edytor

Wygląd edytora jest wzorowany na tym z ZX Spectrum 128. Taka sama jest kolorystyka jak również sposób wpro-



Edytor pobieramy z Internetu...

dzania danych - instrukcje wpisuje się litera po literze. Każdemu naciśnięciu klawisza towarzyszy charakterystyczny dźwięk znany z 48k lub 128k - co jest kwestią konfiguracji. Przy wpisaniu kilku początkowych liter słowa kluczowego pojawia się podpowiedź jego pełnej wersji. Niestety słowo kluczowe trzeba wpisać do końca, gdyż nie ma autouzu-

pełniania. U góry edytora widzimy ikonki, dzięki którym możemy załadować lub zapisać program na dysku, uruchomić go w kilku trybach pracy, dodawać break-pointy, również w kilku wariantach lub też uruchomić podręczny kalkulator wyrażeń.

Na dole okna znajduje się informacja o rodzaju BASICa (48 lub 128), sugestia ty-



pu danych (numer linii, słowo kluczowe lub wyrażenie), jakie należy wprowadzić.

Pomocne we wprowadzaniu programu jest okno nazwane "Token table". Jest to nic innego jak podręczny słownik słów kluczowych, znaków oraz kodów kontrolnych. Aby je wprowadzić należy dwukrotnie kliknąć na wybranym elemencie. Równie pomocna jest tabela "Error help" z wyjaśnionymi kodami błędów oraz jego przyczynami.

Uruchomienie wpisanego programu jest proste. Wystarczy wybrać jedną z ikonek opisanych powyżej, lub też wpisać w edytorze słowo kluczowe:

RUN lub **GOTO** <numer linii>

gdzie numer linii będzie oznaczał tę, od której należy zacząć wykonywać program. Dodam, że RUN przed uruchomieniem czyści wszystkie zmienne zdefiniowane wcześniej i przyjmuje również jako parametr numer linii, GOTO zaś nie czyści zmiennych.

Emulator

Okno emulatora ma kilka ciekawych opcji. Jeżeli chcemy powiększyć je o 100 lub 200% lub też rozciągnąć na dowolny obszar, poprzez przeciągnięcie prawego dolnego rogu okna. Obraz z ekranu emulatora możemy wydrukować dzięki opcji Print.

Więcej możliwości dotyczących okna emulowanego Spectrumba poznamy po



Efekt po uruchomieniu.

zapoznaniu się z jego konfiguracją, w zakładce Display.

Konfiguracja

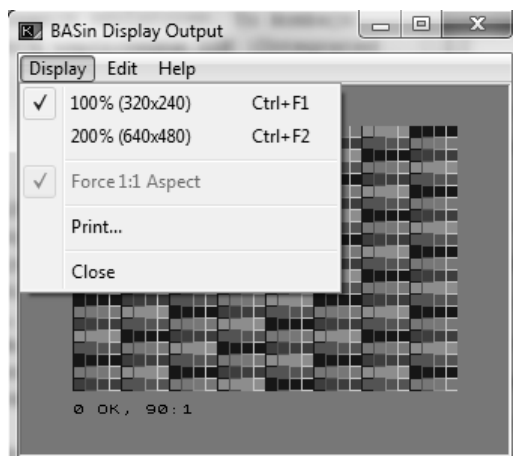
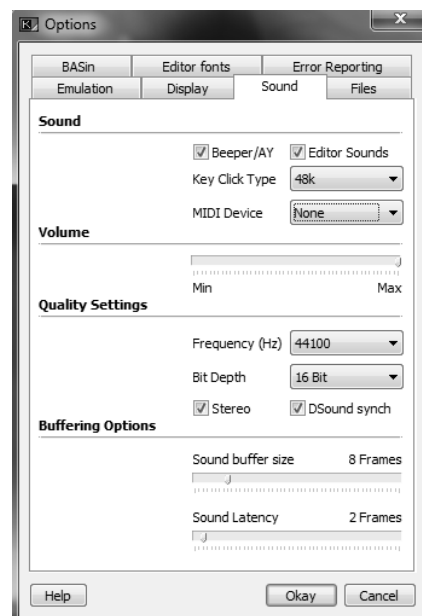
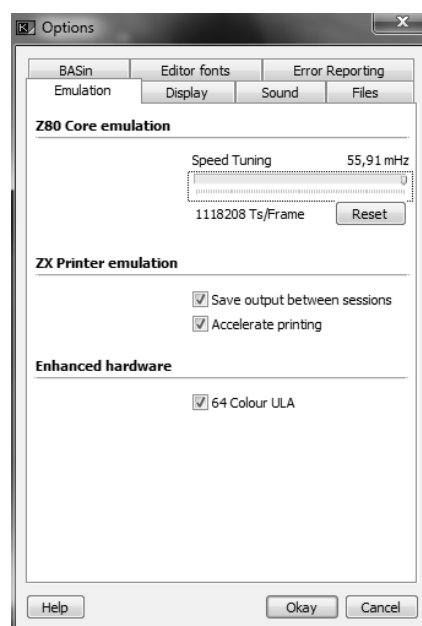
BASin ma rozbudowaną konfigurację, która pozwala indywidualnie ustawić wiele rzeczy. Okno ustawień włącza się poprzez menu kontekstowe i wybiera Tools a następnie BASin options... Oto najistotniejsze z ustawień.

W zakładce "Emulation" znajduje się suwak dzięki któremu ustawimy parametr Speed Tuning. Dzięki niemu ustawiamy szybkość wykonywania programów względem oryginalnej prędkości działania procesora Z80 w ZX Spectrum. Maksymalne przyspieszenie jakie możemy uzyskać to prawie 16-krotność oryginału.

Zakładka "Display" skrywa opcje sterujące wyświetlaniem emulowanego obrazu Spectrum. Tu możemy wybrać sposób skalowania, omijania ramek przy generowaniu obrazu czy też włączenie symulowania obrazu telewizyjnego.

"Sound" to zakładka, w której możemy wybrać rodzaj kliku klawiatury, jego głośność, jak również jakość wydawanego dźwięku przez emulator. W zakładce "Files" ustawimy wersję formatu .z80 jak również dodatkowe nagłówki dla plików tekstowych .bas.

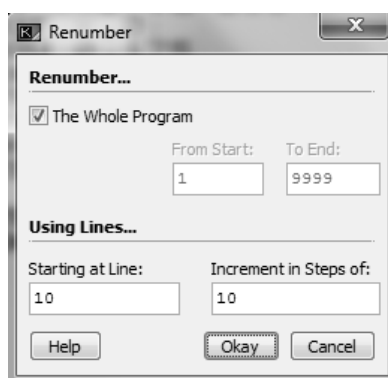
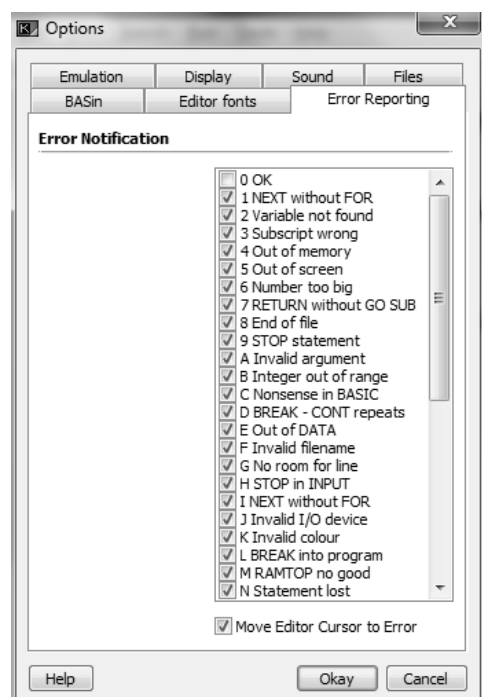
"Error reporting" zawiera listę błędów zgłaszanych przez interpreter BASICA oraz sposób informowania o tych błędach.



Opcje wyświetlania obrazu.



Odznaczenie błędu oznaczać będzie brak notyfikacji. Warto wyłączyć notyfikację dla błędu numer 0 OK oznaczają-



Whole Program" co oznacza, że renumeracja ma dotyczyć całego programu. Jej odznaczenie umożliwi nam wpisanie zakresu linii, jakich dotyczyć ma renumeracja. W polu "Starting at Line:" należy podać numer linii początkowej po renumeracji, a pole "Increment in Steps of:" zawiera informacje o różnicy w kolejnych numerach linii.

W przypadku podania wartości "10" każda nowo powstała linia będzie miała numer o 10 większy od pozostałej.

cą poprawne wykonanie programu. Zamiast okienka z informacją, wyświetlony zostanie komunikat tak jak miało to dotychczas miejsce w ZX Spectrum.

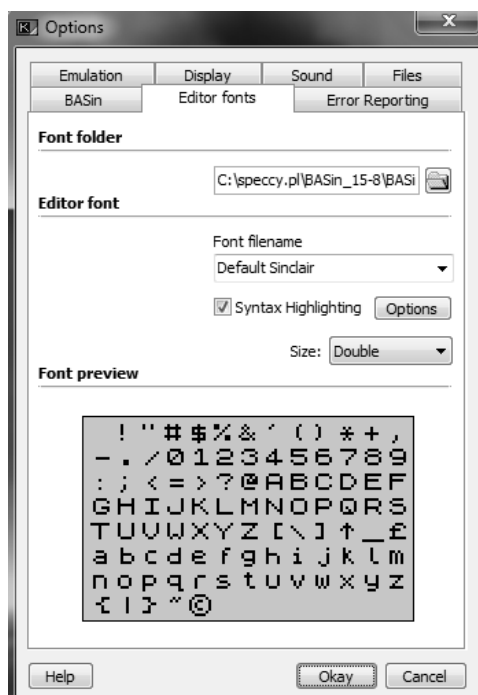
"Editor fonts" umożliwia wybranie czcionki wyświetlanej na ekranie. Jeżeli chcemy wybrać dostarczone z programem czcionki, należy poprawnie wskazać lokalizację katalogu font (opcja Font folder). Po wybraniu prawidłowej ścieżki będziemy mogli wybrać odpowiedni krój liter (Font filename). Dzięki opcji "Size" możemy zwiększać rozmiar czcionki użytej w edytorze. Mamy również możliwość konfiguracji podświetlania składni.

Zakładka 'BASin' zawiera ogólne ustawienia.

Renumeracja linii

Każda linijka programu w BASICu ma swój numer. O ile przy małych programach nie stanowi to problemu, tak przy dużych, które zostają wielokrotnie rozbudowywane, stanowi to pewne wyzwanie. Nowo powstałe linie programu trzeba 'wcisnąć' między już istniejące i tu z pomocą przychodzi nam bardzo pożyteczna funkcja - Renumeracja. Wywołuje ją się poprzez menu kontekstowe Tools -> Renumber...

Jego obsługa jest bardzo prosta. Standardowo zaznaczona jest opcja "The



W kolejnej części artykułu przedstawię kilka modułów, między innymi rozbudowany debugger oraz służących do edycji grafiki. Poruszę również temat tworzenia plików .tap z napisanych programów oraz danych jak również zdemonstruję możliwość pisania własnych wstawek w assemblerze, które będzie można wykorzystać z programach BASICowych.

Krystian "Tygrys" Włosek

W NASTĘPNYM NUMERZE

SINCLAIR ZX SPECTRUM VEGA



Nowy numer już w maju!

RGB

Metody konwersji sygnału

Czy decyzja o wyprowadzeniu wyłącznie RGB jako sygnału wizyjnego z komputera marki Amstrad należała do Sir Alana Michaela Sugara? Prawdopodobnie tak. Jest on przecież twórcą i byłym właścicielem tejże firmy, która produkowała popularne w latach 80. komputery domowe z serii Amstrad CPC. Sprzedawane od 1984 roku modele 464, 664 i 6128, a także późniejsze z serii plus, z wyłączeniem konsoli GX4000, były pozbawione kompozytowego wyjścia wideo.

Każdy z tych komputerów sprzedawany był w komplecie z monitorem. Modele GT64/65 wyświetlały monochromatyczny obraz o zielonej barwie, zaś CTM640/644 pozwalały na pracę w kolorze. Właśnie te ostatnie przystosowane były wyłącznie do odbioru sygnału RGB. Po co firma miałaby podrażać koszty i montować w komputerach dodatkowe układy konwertujące lub modulatory antenowe i dawać użytkownikom swobodę w podłączaniu ich urządzeń pod dowolne odbiorniki? Głosząc ideę - „Po otwarciu pudełka nie potrzebujesz nic więcej”, sprzedawała użytkownikowi komputer wraz z monitorem zakładając, że telewizor w salonie nie będzie musiał być odbiornikiem ich urządzenia. Pomysł ciekawy, ale po latach, kiedy potrzeby użytkowników takich jak ja, nie do końca pokrywają się z koncepcją decydentów firmy Amstrad, wydaje się być chybiony.

Dzięki takiej polityce, stanąłem ostatnio przed problemem podłączenia kompu-

tera Schneider CPC 6128 do współczesnego projektora multimedialnego, na którym miały być wyświetlone demo. Okazało się, że projektor wspiera wyłącznie sygnały S-VHS, kompozyt wideo i VGA. Próbując spiąć razem nowoczesne urządzenie z wiekowym komputerem, bezradnie rozłożyłem ręce. Ziarenko zostało jednak zasiane i postanowiłem, że przy następnej okazji (jak się później okazało był to Retro-Komp/LOAD ERROR 2015), będę w pełni gotowy, by pokazać co ten komputer ma do zaoferowania.

Zaczęło się przeszukiwanie przepastnych zasobów Internetu, pisanie na forach i analizowanie schematów. Okazało się, że wyjść z sytuacji jest kilka. Niektóre wymagały większych nakładów finansowych, inne - mniejszych, ale były trudniejsze do wykonania. W tym artykule opiszę moje doświadczenia i znane mi metody konwersji sygnału RGB do innych sygnałów wideo. Przy tym większość z nich może być zastosowana nie tylko w serii komputerów marki Amstrad.

Metoda pierwsza:
Mamy dużo pieniędzy i dwie lewe ręce do lutowania.

Dobłą, aczkolwiek drogą alternatywą, jest zakup urządzenia do konwersji sygnału RGB. Profesjonalne konwertery (np. GBS-8219 firmy Gonbes) do zasto-



Konwerter sygnału RGB do S-Video.

sowań przemysłowych (monitory obrabiarek CNC, wtryskarek itd.) przetwarzają sygnały RGB, YPbPr i inne do sygnału VGA. Ich cena jest jednak stosunkowo wysoka dlatego nie będę się o nich rozpisywał. Do moich potrzeb wystarczy urządzenie firmy Keene Electronics, o nazwie RGB TO S-VIDEO CONVERTOR. Jego możliwości i domniemana dobra jakość wykonania wydają się rekompensować niemałą cenę.

Na dzień dzisiejszy wynosi ona po przeliczeniu 360 PLN. Urządzenie zamknięte jest w zgrabnej, niewielkiej obudowie i jak wynika z opisu producenta i zdjęć katalogowych, konwersja możliwa jest zarówno do sygnału S-VHS, jak i kompozyt wideo. W konwerterze jest też wyjście audio stereo, gdyby okazało się, że w podłączanej do urządzenia wtyczce SCART taki sygnał się znajduje i musi zostać przepuszczony dalej.



Konwerter wymaga osobnego zasilania i jak zapewnia producent, automatycznie wykrywa, czy podłączony do wejścia sygnał jest w standardzie PAL czy NTSC oraz odpowiednio się do niego dostosowuje. Z dalszego opisu wynika, iż wewnątrz znajduje się mikrokontroler, który zapewnia doskonałą stabilność obrazu. W tym momencie mogę jedynie zaufać zapewnieniom producenta, ponieważ nie było mi dane sprawdzić działania tego urządzenia w praktyce. Trochę szkoda, ale poszedłem inną, mniej kosztowną drogą. Być może przy następnej okazji potwierdzę zapewnie-

nie wyjście wideo w jednej z pierwszych serii ich 16-bitowych komputerów, między innymi w Amidzie 500. Praktycznie uniemożliwiało to pracę na odbiorniku innym niż monitor, który był w stanie przyjąć sygnał RGB. Może powodem było cięcie kosztów (konieczność montowania w komputerze dodatkowego enkodera sygnału wideo), a może wymuszenie na użytkownikach zakupu firmowego monitora. To rozważania, które nie są celem tego artykułu.

Faktem jest, że wkrótce znalazła się na to rada, bo producent rzeczonych komputerów wypuścił na rynek w 1988 roku modulator o A520. Urządzenie to, oparte na układzie scalonym Motorola MC1377,

Opis przeróbki, z jakiej korzystałem, można znaleźć na stronie internetowej pana Krzysztofa Komarnickiego pod adresem:

sites.google.com/site/krzkomar/krzkomar_e-1_pl/modulator-amiga-a520

W skrócie polegała ona na usunięciu części elementów elektronicznych przetwarzających sygnał zegara _CCK, którego to inne komputery niż Amiga nie posiadają i podłączeniu w to miejsce rezonatora kwarcowego oraz kondensatora nastawnego, którym regulujemy starając się uzyskać kolor na ekranie. Konieczne było także użycie zewnętrznego zasilacza 12VDC, ponieważ mój Schneider CPC zasilany jest wyłącznie napięciem 5VDC (trzycałowa stacja dysków, która wymaga zasilania 12VDC, jest w moim komputerze odłączona).



Modulator Commodore A520.

nia firmy Keene, dokładnie testując opisany konwerter.

Należy jeszcze dodać, że przy opcji skorzystania z tego urządzenia, konieczny będzie zakup specjalnego kabla, z jednej strony zakończonego 6-pinowym wtykiem typu DIN (który wpinamy do gniazda komputera), a z drugiej strony wtykiem męskim typu SCART. Wymagane przez konwerter sygnały wychodzące z gniazda RGB Amstrada CPC to trzy składowe kolory: czerwonego, zielonego i niebieskiego, sygnał synchronizacji oraz masa. Takie kable są dostępne między innymi na zagranicznych portalach aukcyjnych, takich jak:

www.ebay.co.uk

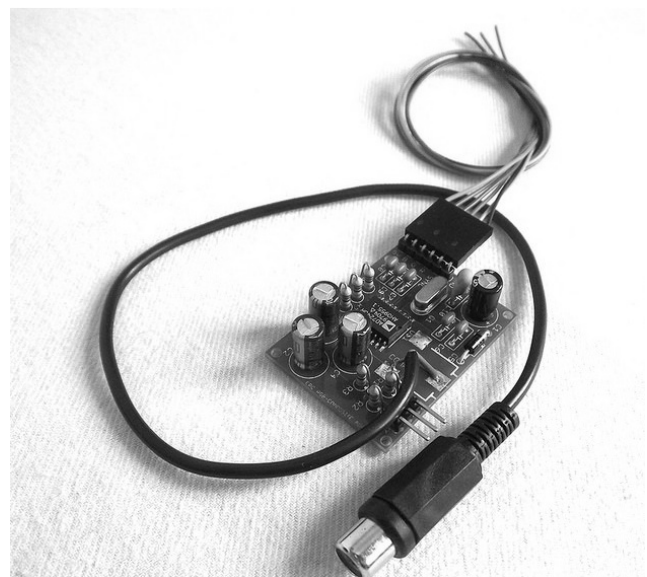
Metoda druga:
Lubimy majsterkować i jesteśmy w posiadaniu modulatora A520 do komputera Amiga 500.

Z jakiegoś powodu firma Commodore zdecydowała się na monochromatycz-

po podpięciu na przykład do Amigi 500, umożliwiałoby wyświetlenie kolorowego obrazu na wielu różnych odbiornikach. Modulator A520 generuje bowiem kolorowy sygnał kompozyt wideo, a także, poprzez wbudowany modulator UHF, sygnał antenowy.

Obecnie zdobycie używanego i sprawnego egzemplarza nie powinno stanowić problemu. Jego koszt może wahać się w granicach 20-40 PLN w zależności od stanu. Sama przeróbka nie jest skomplikowana ani kosztowna i wymaga jedynie odrobiny cierpliwości oraz podstawowej wiedzy z zakresu techniki lutowania.

Wymagane jest także posiadanie kilku części, złączy (w tym trudno dostępnego DB-23, które z powodzeniem można zastąpić przerobionym złączem DB-25) i zewnętrznego zasilacza, który doprowadzi napięcie 12VDC do enkodera MC1377.



Konwerter oparty na układzie AD724JR.

Cały układ po zmontowaniu nie był zbyt poręczny i wygodny w użytkowaniu. Regulacja trymerem była konieczna przy prawie każdorazowym uruchomieniu zestawu, co uniemożliwiało wygodną obsługę. Obraz testowałem na kilku monitorach. Początkowo był on marnej jakości, nawet jak na kompozyt wideo i miałem wrażenie, iż po złączeniu antenowym jest o niebo lepszy. Podczas testów udało mi się go poprawić i trochę zniwelować te zakłócenia. W tym celu musiałem zrezygnować z fabrycznego wyjścia kompozytowego wideo modulatora A520 i pobrać wymagany sygnał z innego miejsca, a dokładniej sprzed wewnętrznego modulatora UHF. Być może była to przypadłość tego konkretnie egzemplarza.

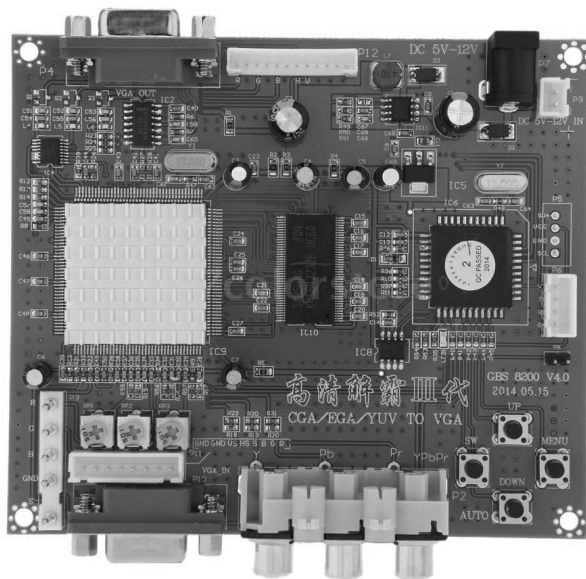


Kolejnym problemem okazała się niemożność wykorzystania tego rozwiązania w komputerze Atari XL/XE z rozszerzeniem VBXE. Z niewiadomych przyczyn brakowało kolorów, a obraz tracił synchronizację. Na koniec należy dodać, że istnieje możliwość zbudowania własnego układu konwertującego opartego na Motoroli MC1377. Korzystniej jednak będzie zrobić to na układzie bardziej nowoczesnym, o czym napiszę w kolejnym punkcie.

Metoda trzecia:
Lutownica nam nie straszna,
a technika SMD to bułka
z masłem.

W przeciwieństwie do układu scalonego MC1377, jego nowszy „odpowiednik” AD724JR (przeznaczony wyłącznie do montażu powierzchniowego) wymaga jedynie zasilania 5VDC. Daje to możliwość wykonania opartej na nim płytki, która może być zamontowana wewnątrz komputera, który nie posiada napięcia 12VDC. Jakby tego było mało, ów układ scalony na wyjściu daje także sygnał S-VHS a nie tylko, jak opisany wcześniej modulator A520, kompozyt video.

Korzystając z informacji zawartych na stronie:



Konwerter GBS-8200.

www.cpcwiki.eu/index.php/RGB_SVideo

można zbudować własną płytkę elektroniczną, która bez problemu wpasuje się w obudowę Amstrada CPC. Na stro-

nie znajdziemy dokładny opis, schematy i szablony do płytek drukowanych. Rozwiązanie zaproponowane przez użytkownika o pseudonimie Bryce, oparte zresztą na oryginalnej karcie charakterystyki producenta układu AD724JR, jest eleganckie i kompaktowe. Przy założeniu, że mamy już gotową płytkę, jej instalacja jest dziecinnie prosta i sprowadza się do polutowania kilku przewodów w okolicy wyjścia monitorowego Amstrada CPC i wykonania otworu w obudowie, na dodatkowe gniazdo kompozyt video i/lub S-VHS.

O ile duża dostępność i niska cena elementów pomocniczych proponowanego układu jest bezdyskusyjna, to już zakup AD724JR w Polsce w dobrej cenie nie należał do łatwych zadań. Mnie udało się go nabyć u chińskiego dystrybutora (ponownie eBay) za bardzo przystępną cenę 10 PLN za sztukę. W polskim sklepie z częściami elektronicznymi znalazłem taki sam za ponad 50 PLN. Niestety brak możliwości profesjonalnego wykonania płytki zmusił mnie do zbudowania jedynie testowej wersji tego układu.

Całość została wykonana na płycie uniwersalnej z wykorzystaniem elementów do

www.ebay.co.uk

(tak, tak, znowu on) i udało mi się nawiązać kontakt z użytkownikiem 8bits4ever, który montuje takie układy profesjonalnie (solidne wykonanie, dołączona instrukcja montażu itd.) i w przystępnej cenie 70 PLN. Na dzień dzisiejszy jest to rozwiązanie, z którego korzystam.

Metoda czwarta:
Kupujemy tani konwerter od
Chińczyka.

Celowo nie wrzuciłem tego rozwiązania do jednego worka wraz z pozostałymi konwerterami omówionymi w pierwszym punkcie artykułu. Urządzenie, które będzie tu opisywane, aby zadziałało w zasadzie nie wymaga specjalnych przygotowań, ale też do końca nie jest rozwiązaniem dla osoby nie dysponującej podstawową wiedzą o technice lutowania czy odpowiednimi narzędziami.

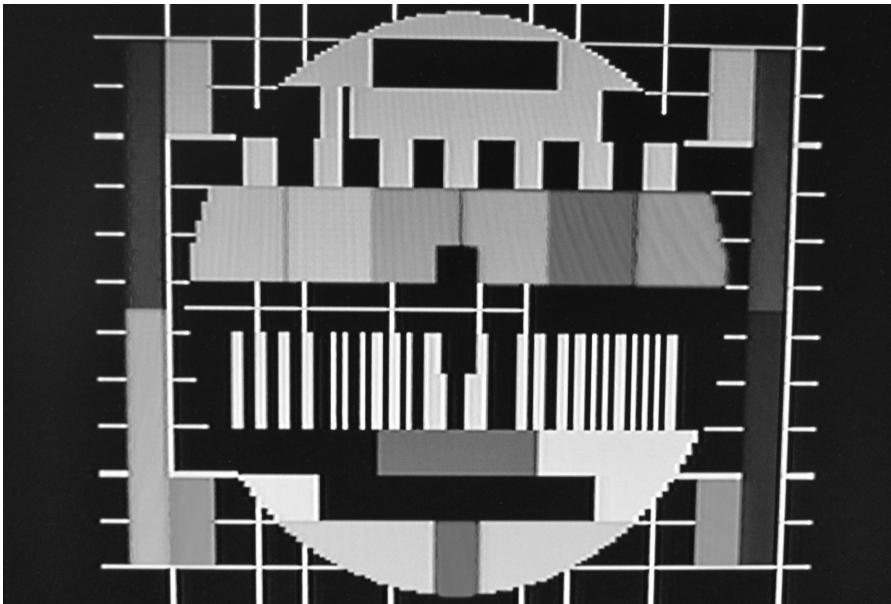


Modulator MP-1.

montażu przewlekane-go. Mimo tak przygotowanej wersji, efekt był piorunujący. Obraz bardzo ostry, kolory kompletne i nie przekłamane i żadnych zakłóceń na miarę modulatora A520. Oczywiście docelowo chciałem zbudować lub zlecić budowę układu z prawdziwego zdarzenia. Moja pajęczyna przewodów w każ-dej chwili mogła zawieść.

Na szczęście z pomocą przyszedł:

Konwerter GBS8200 (zwany także HD9800) to w zasadzie, jak podpowiada nazwa, kopia urządzenia do zastosowań przemysłowych, opisywanego na początku tego artykułu. Jego cena jest jednak o wiele, wiele niższa. Za to kupujemy najprawdopodobniej podróbkę, opartą zapewne na tanich podzespołach, bez obudowy, zasilacza i kabli sygnałowych. Układ wyświetla obraz po złączu VGA. Bez problemu podłączymy więc do niego prawie każdy współczesny monitor LCD, a także starsze CRT, projektor multimedialny czy w końcu telewizor z odpowiednim złączem.



Obraz kontrolny z A520.

Niestety, z racji braku droższego egzemplarza, nie byłem w stanie porównać jakości wyświetlanego obrazu obu tych urządzeń. Mogę jedynie zapewnić, że GBS8200 to dobre rozwiązanie przy uwzględnieniu stosunku ceny do jakości. Obraz generowany przez niego miał tylko niewielkie zakłócenia w postaci pojawiających się białych punktów na ekranie, ale wina najpewniej leżała po stronie kabla VGA. Problemem było też pozycjonowanie obrazu na ekranie telewizora, ale to jednorazowy zabieg.

Konwerter działa także z ośmiobitowym Atari XL/XE wyposażonym w rozszerzenie VBXE, co dla wielu osób może

okazać się niezwykle ważne. Urządzenie przyjmuje istotny dla nas sygnał RGB, a także CGA/EGA czy YPbPr. Właśnie w przypadku RGB i CGA/EGA konieczne jest przygotowanie specjalnego kabla, którego w sprzedaży raczej nie znajdziemy. Podobnie jak w przypadku konwertera firmy Keene konieczne jest dostarczenie do urządzenia czterech sygnałów oraz masy. Pozostaje też problem jak zabezpieczyć płytkę przed uszkodzeniami. Jest zbyt duża, żeby zmieścić ją wewnątrz komputera, jednocześnie brak obudowy naraża ją na uszkodzenia mechaniczne oraz większy kontakt z pyłem czy przypadkowymi zalaniem. Stosowanie osobnego zasilacza o napięciu 5VDC też może

być uciążliwe, chyba że podepnimy się pod zasilanie komputera.

Jakkolwiek by na to nie patrzeć, uniwersalność konwertera GBS8200, a także szeroka gama odbiorników ze złączem VGA, pod jakie możemy podpiąć urządzenie, może zadecydować o wyborze właśnie tego rozwiązania.

Metoda piąta: Przerabiamy fabryczny modulator Amstrada MP-1/MP-2.

Po decyzji firmy Amstrad polegającej na rezygnacji z fabrycznego wyjścia kompozytowego wideo z komputera z serii CPC, na rynek w latach świetności tego komputera trafiło urządzenie o nazwie MP-1/MP-2. Było to idealne rozwiązanie dla tych, którzy posiadali jedynie monochromatyczny monitor GT64/65 i chcieli użytkować komputer w kolorze, nie mając jednocześnie wystarczających środków na zakup fabrycznego, kolorowego monitora.

Urządzenie pozwalało na podpięcie jedynie odbiornika telewizyjnego z wejściem antenowym UHF. Wewnątrz modułu, poza wbudowanym zasilaczem, znajduje się konwerter sygnału RGB oparty podobnie jak w A520 na układzie scalonym MC1377. Istnieje przeróbka, która umożliwia uzyskanie obrazu „po kompozycie”, podobnie jak w przypadku modulatora A520. Jest ona jednak o wiele łatwiejsza i sprowadza się do jednej prostej operacji. Wystarczy przylutować dwa przewody między dwa pola lutownicze na płycie modulatora a żeńskie gniazdo RCA, które możemy zamontować w obudowie (po wcześniejszym przygotowaniu otworu) lub przewód, którego koniec jest męskim wtykiem RCA, podpinanym bezpośrednio do telewizora (co oszczędzi nam niszczenia obudowy).

Według zebranych przeze mnie informacji, sygnał jaki daje ta przeróbka jest nie najgorszy. Jednak z tego, co udało mi się zaobserwować na zdjęciach, wygląda gorzej niż po przeróbce modulatora A520. Sam konwerter MP-1/MP-2, mimo iż jest większy i cięższy od modulatora do Amigi, wydaje się być bardziej mobilny. Nie potrzebujemy już dodatkowego zasilacza, ponieważ jest on w nim wbudowany. Konwerter wystarczy podpiąć do gniazdzka sieciowego, podłączyć zasilanie do komputera Amstrad CPC i sygnał wideo do telewizora lub projektora z przerobionego



Demo z AD724JR.



Gniazdo S-VHS wyprowadzone z CPC.

Powiązane linki:

www.cpcwiki.eu/index.php/RGB_SVideo

www.speccy.pl

sites.google.com/site/krzkomar/krzkomar_el_pl/modulator-amiga-a520

pl.wikipedia.org

www.cpcwiki.eu/index.php/Amstrad_MP1/MP2_modulator

atariki.krap.pl

przez nas wyjścia. Niestety na polskich portalach aukcyjnych ten konwerter bardzo rzadko się pojawia, w przeciwieństwie do popularniejszego modulatora A520.

Istnieje jeszcze szóstą metodą, którą trudno określić jako konkurencyjną dla pozostałych. Dzięki niej można bowiem uzyskać jedynie czarno-biały obraz. Wystarczy, że połączymy przy pomocy kabla RCA z uciętym z jednej strony wtykiem, dwa piny gniazda monitorowego komputera Amstrad z kompozytowym wejściem wideo w telewizorze. Na jednym z tych pinów (szóstym w modelach CPC 464, 664 i 6128), znajduje się sygnał luminancji. Piąty pin to masa.

Metoda ta będzie odpowiednia chyba jedynie przy testowaniu świeżo nabytego komputera, określonego często mianem „niesprawdzony”, gdy nie będziemy jeszcze dysponować odpowiednim okablowaniem czy fabrycznym monitorem.

Podsumowując, wybór pierwszej metody (zakup profesjonalnego konwertera) będzie właściwy raczej dla kogoś, kto zechce używać tego modułu z wieloma komputerami i to w różnych, czasem "bojowych" warunkach (złoty, party itd.). Wtedy warto zainwestować. Można oczywiście zakupić tańszą podróbkę (metoda czwarta), jeśli komuś nie przeszkadza

widok "gołej" płytki i gdy komputer będzie używany w jednym miejscu, na przykład w domu. Najlepiej wówczas ukryć ją gdzieś za telewizorem czy monitorem (przypominam, że tu uzyskamy jedynie sygnał VGA). Z kolei przerobienie modulatora A520 (metoda druga) nie jest według mnie najlepszym rozwiązaniem mimo, iż może się okazać najtańszym dla kogoś, kto taki modulator już posiada. Najkorzystniej wypada zakup lub wykonanie płytki opartej na układzie AD724JR (metoda trzecia). Montaż wewnątrz komputera zapewni brak konieczności stosowania dodatkowych zasilaczy i zagwarantuje minimalistyczny wygląd zestawu, z możliwością szybkiego i uniwersalnego podłączenia pod prawie każdy odbiornik.

Wówczas do wyboru będzie fabryczny sygnał RGB, a także S-VHS i kompozyt wideo. Jednakże, przy korzystaniu z kilku różnych komputerów (np. modele klasyczne i te z serii plus), konieczne będzie zamontowanie po jednej płytce w każdym z nich.

Michał "stRing" Radecki-Mikulicz

Podziękowania dla:

Yolk, Krzysztof Komarnicki, Bryce, speccy.pl, cpcwiki.eu.

Przypisy:

Wyższy numer w typie monitora do Amstrada CPC (w tym wypadku GT65 i CTM644) oznacza, iż posiada on dodatkowe wyjście zasilania do wewnętrznej, trzy calowej stacji dysków, która do działania potrzebowała 12VDC.

Komputery firmy Amstrad były produkowane i sprzedawane w Niemczech, Austrii i Szwajcarii przez firmę Schneider Computer Division (cóрка firmy Schneider Rundfunkwerke AG) jako Schneider CPC. Schneider Rundfunkwerke AG ogłosiło w 2002 roku bankructwo.

Video Board XE, w skrócie VBXE, to rozszerzenie sprzętowe autorstwa Electrona, przeznaczone dla ośmiobitowego Atari. Polega na instalacji układu, który naśladując działanie GTIA, generuje sygnał wizyjny RGB.

Różnica pomiędzy modulatorem MP-1 i MP-2, podobnie jak w przypadku monitorów do Amstrada, polegała na obecności dodatkowego zasilania 12VDC. Modulator w wersji MP-1 dostarczał jedynie 5VDC.

Istnieje jeszcze urządzenie o nazwie MP-3, które jest także modulatorem, ale działającym niejako "w drugą stronę". Umożliwia ono bowiem wykorzystanie fabrycznego monitora (np. CTM644) jako odbiornika sygnału telewizji analogowej.



Interfejs SatanDisk

Jako zagorzały fan produktów marki Commodore, aż do 2014 roku byłem zupełnym ignorantem w kwestii komputerów Atari. Powoli zaczęło się to zmieniać, kiedy zapoznałem się z wieloma produkcjami atarowskiej demosceny podczas kolejnych edycji gdańskiego RetroKompa oraz zainteresowałem się udziałem w największym atarowskim party scenowym – Silly Venture. Odczuwałem przemożną chęć wzbogacenia swojej kolekcji komputerów o 16-bitowe Atari.

Mój wybór padł na model STe, który pozwala na uruchomienie wszystkich demoscenowych produkcji na duże Atari (oczywiście nie licząc Falcona) i w dość krótkim czasie nabyłem model 520 STe, z 4 MB pamięci RAM i TOS-em 2.06. Po zakupie komputera zacząłem rozglądać się za jakimś nowoczesnym rozwiązaniem, pozwalającym na korzystanie ze współczesnych nośników pamięci. Jako, że po zakupie samego komputera z funduszami zrobiło się kruczo i nie mogłem sobie pozwolić na nabycie najbardziej polecanego urządzenia - UltraSatan, produkowanego przez Lotharka, mój wybór padł na tańsze i nieco starsze rozwiązanie, czyli interfejs SatanDisk.

Interfejs znalazłem na portalu Allegro, sprzedawał go użytkownik Darek324, czyli Dariusz Pyciak będący jednocześnie producentem SatanDiska. Cena była przystępna i wynosiła około 100 zł, nie licząc przesyłki.

Więcej informacji o urządzeniu, a także kontakt z samym producentem znajdziemy na jego stronie internetowej pod poniższym adresem:

toconasze.pl

Czym jest interfejs SatanDisk?

SatanDisk jest urządzeniem pozwalającym na użycie w „dużych” Atari kart SD/MMC jako pamięci masowej, widocznej w systemie jako zwykły twardy dysk. Urządzenie zostało zaprojektowane w roku 2007 przez Słowaka Miroslava Nohaja, znanego także jako Jookie, który udostępnił projekt na zasadzie Open-Source.

SatanDisk obsługuje karty o pojemności do 4GB i działa z systemami TOS w wersjach od 1.02 do 2.06 (z ograniczeniami można go uruchomić nawet na TOSie 1.00 - przyp. Red.). Interfejs kompatybilny jest ze sterownikami HDDRIVER oraz oprogramowaniem przygotowanym przez Pera Putnika.

Pierwsze wrażenie i montaż

SatanDisk jest niewielkim urządzeniem o wymiarach 56x48mm. Poza samą

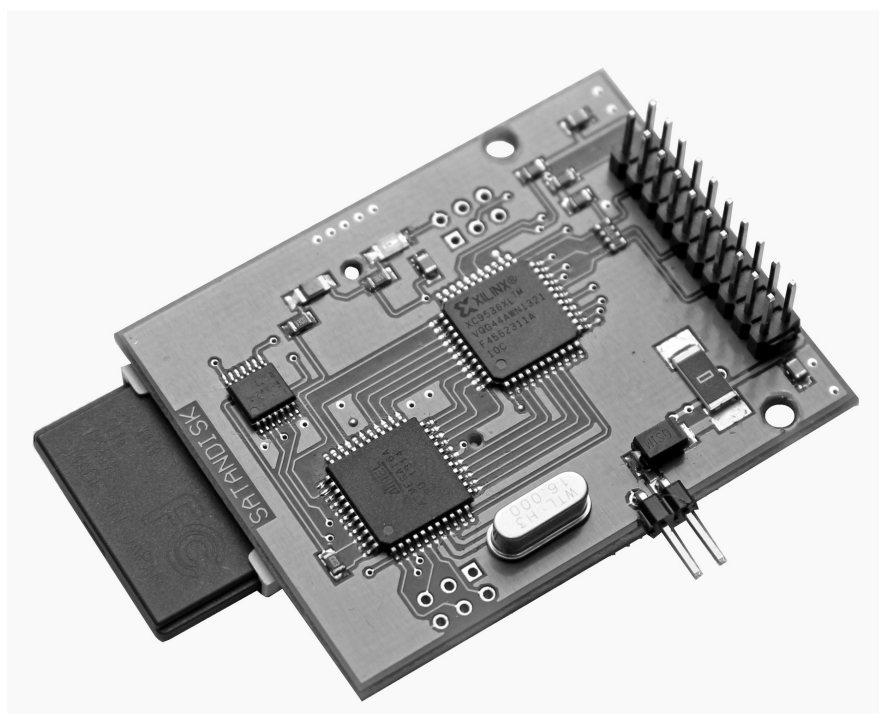
płytką drukowaną, w zestawie otrzymałem taśmę sygnałową do połączenia z Atari ST, zasilacz 12V DC oraz płytkę CD-R z oprogramowaniem i instrukcją w formacie PDF.

Pierwszą rzeczą, na jaką zwróciłem uwagę była wtyczka do podłączenia interfejsu do komputera. Gniazdo Hard Disk w Atari ST ma 19 pinów, natomiast wtyczka w dostarczonym urządzeniu była przerobiona (poprzez usunięcie 6 ostatnich pinów) z wersji 25-pinowej.

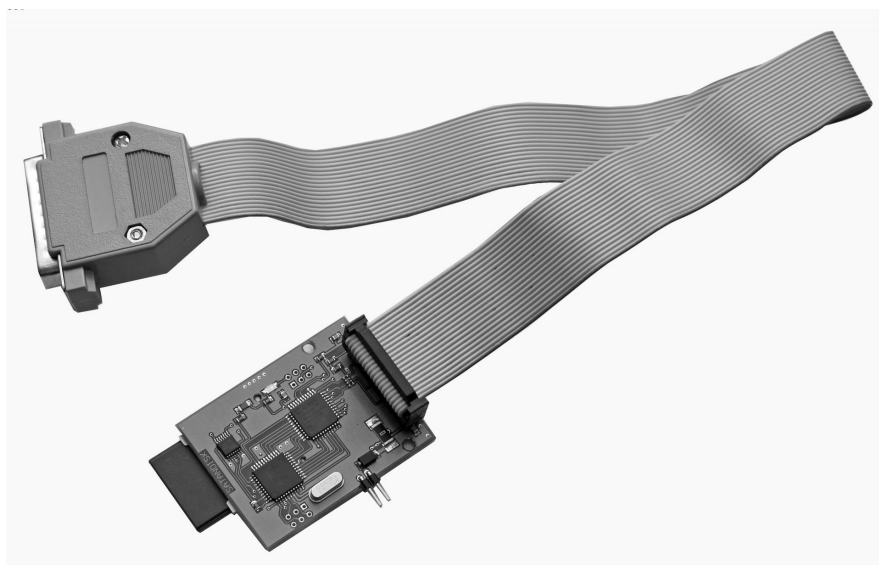
Jako, że wtyczka jest nieco szersza niż gniazdo, należy przed montażem usunąć tulejkę dystansową z gniazda Hard Disk, może też być wymagane lekkie podpiłowanie plastiku przy gnieździe, bądź przeróbka samej wtyczki. Z uwagi na to, że nie chciałem ingerować w obudowę mojego nowo zakupionego Atari, postanowiłem przerobić wtyczkę, do czego wystarczyły zwykłe obcążki.

Urządzenie dostarczane jest bez obudowy, jednak producent na swojej stronie umieścił zdjęcia dedykowanej niewielkiej obudowy, którą można wydrukować sobie na drukarce 3D. Urządzenie można także, po odpowiedniej modyfikacji obudowy, zamontować wewnątrz naszego Atari, pobierając napięcie z zasilacza komputera.

Po podłączeniu interfejsu do Atari ST, pozostaje umieszczenie w nim przygotowanej uprzednio karty SD/MMC i podłączenie zasilania.



Płytkę interfejsu z kartą pamięci SD.



Komplet z przewodem połączeniowym.

Użytkowanie SatanDiska

Przed rozpoczęciem pracy należy przygotować kartę SD/MMC, którą chcemy przeznaczyć do działania z naszym Atari. Na dołączonej do urządzenia płycie znajdują się między innymi: demonstracyjna wersja wspomnianego już sterownika HDDRIVER oraz obrazy kart z zainstalowanym driverem P. Putnika, które przeznaczone są dla kart o różnej wielkości. Obraz karty zawierający zainstalowany sterownik, a także wiele nagranych gier można pobrać bezpośrednio ze strony:

<http://atari.8bitchip.info/DiskImgPP1.html>

Proces nagrywania obrazów na kartę nie jest skomplikowany i wymaga użycia windowsowego programu Drive Image, nagranych na dołączonej do SatanDiska płycie. Po poprawnym nagraniu na kartę obrazu - również z dołączonej płyty, bądź ściągniętego ze strony Pera Putnika, system Windows wykrywa pierwszą z partycji, którą możemy wykorzystać do transferu danych. Uzyskanie dostępu do pozostałych partycji pod systemami Microsoftu wymaga nieco więcej gimnastyki i zmuszenia systemu do traktowania karty SD/MMC jak dysku twardego.

Bez problemu możemy też przygotować kartę do użytku z SatanDiskiem korzystając z Linuksa. Cały proces został dokładnie opisany na stronie nieocenionego Pera Putnika:

<http://atari.8bitchip.info/drimlus.php>

Niestety nie każda karta bezproblemowo zadziała z Atari. W moim przypadku na dwie sprawdzone karty, z nagraniem tym samym obrazem dysku, jedna zadziałała, a druga już nie.

Przygotowaną kartę umieszczamy w gnieździe SatanDiska przed podłączeniem zasilania do interfejsu. Urządzenie należy włączyć przed uruchomieniem Atari, inaczej nie będzie widoczne w systemie. Po podłączeniu zasilania powinna zaświecić się zielona dioda na interfejsie oraz na około dwie sekundy - dioda czerwona. Jak mówi instrukcja, jeśli czerwona dioda nie gaśnie, oznacza to błąd komunikacji z kartą SD/MMC i należy wyłączyć i włączyć urządzenie ponownie.



Po uruchomieniu Atari, SatanDisk nie jest od razu wykrywany, należy poczekać na załadowanie się GEM-a i zresetować komputer, dopiero po resecie interfejs zostaje wykryty i mamy dostęp do partycji na karcie. Jest to niestety cecha samej konstrukcji urządzenia, o której pisze także jego projektant Joogie. Kolejnym ograniczeniem konstruk-

cji SatanDiska jest brak możliwości odłączania, oraz zamiany karty podczas pracy urządzenia. Efektem tego jest konieczność wyłączenia interfejsu i samego komputera, kiedy chcemy np. dograć na kartę świeżo ściągnięty soft, lub skorzystać z innej karty.

Używając SatanDiska nie powinniśmy także spodziewać się piorunującej prędkości transferu danych. Według wpisów na forach dyskusyjnych, oraz informacji na stronie konstruktora karty, odczyt danych przebiega z prędkością około 150 KB/s.

Pamiętając o ograniczeniach urządzenia, należy jednak powiedzieć, że bardzo usprawnia ono pracę z Atari ST. Przede wszystkim ogranicza konieczność stosowania dyskietek i bardzo ułatwia wymianę danych ze światem zewnętrznym. Niska prędkość transferu danych także nie dyskwalifikuje urządzenia, ponieważ i tak jest prawie 4-krotnie szybciej niż przy odczycie danych ze stacji dysków, a większość gier i dem uruchamia się w kilka sekund. Niebagatelną zaletą SatanDiska jest także jego niewielki rozmiar, pozwalający chociażby na montaż urządzenia wewnątrz obudowy Atari, chociaż wymaga to trochę pracy.

Podczas kilku miesięcy użytkowania SatanDiska nie napotkałem żadnych problemów ze stabilnością systemu, czy kompatybilnością z grami dostosowanymi do pracy z twardym dyskiem, które są w całych setkach dostępne na stronie P. Putnika. Bez problemu działa także przytłaczająca większość produkcji scenowych. Problem pojawił się z kilkoma nowszymi demami na Atari STe, takimi jak Sea of Colour, ATM 10 Years oraz Drone. Dema te intensywnie streamują z dysku grafikę i dźwięk, więc transfer jaki osiąga SatanDisk jest po prostu za niski.

Podsumowanie

Tak jak napisałem w powyższych akapitach, interfejs SatanDisk nie jest pozbawiony pewnych wad. Wynikają one głównie z tego, że jest to konstrukcja już 8-letnia, a na rynku już od pewnego czasu dostępny jest jej następca, zaprojektowany także przez Joogie'go i produkowany w Polsce przez Lotharka - UltraSatan.

W porównaniu ze starszym „bratem”, UltraSatan oferuje znacznie wyższy transfer danych (do 1 MB/s), możliwość równoczesnego podłączenia dwóch kart i ich wymiany w trakcie pracy, działa także od razu, bez konieczności do-



datkowego resetu. Za nowsze i bardziej skomplikowane urządzenie, jakim jest UltraSatan zapłacimy niestety kilkakrotnie więcej niż za SatanDiska. Wybór tak naprawdę sprowadza się więc nie tyle do funkcjonalności, co do kwoty jaką jesteśmy w stanie zainwestować w swoje duże Atari.

Dla mnie na chwilę obecną, SatanDisk jest wystarczający do tego, żeby raz na jakiś czas włączyć Atari, obejrzeć demo i odpalić grę. Fakt, że nie działa na nim kilka najnowszych produkcji jest irytujący, ale na razie nie na tyle, żeby wymienić urządzenie na droższe. SatanDisk

jest sprzętem tanim i niezawodnym, jego ograniczenia na pewno są zauważalne, jednak w mojej ocenie nie przeszkadzają w pracy w takim stopniu, żeby dyskwalifikować to urządzenie.

Zdjęcia pochodzą ze strony:

<http://toconasze.pl>

wykorzystane za zgodą autora.

Mateusz „Tfardy” Eckert

SatanDisk

Zalety:

- mały rozmiar
- niezawodność
- niska cena

Wady:

- wolny transfer
- konieczność ponownego włączenia komputera i SatanDiska po zmianie karty lub dograniu danych
- konieczność resetu po uruchomieniu komputera

W 1989 roku Atari wypuściło na rynek pierwszą na świecie przenośną konsolę z kolorowym wyświetlaczem. W 1991 pojawiła się druga wersja tej konsoli. Zmniejszono pobór mocy, co pozwoliło na dłuższą zabawę na bateriach, dodano dźwięk stereo oraz zmieniono obudowę. W 2015 roku powstała kolejna modyfikacja, teraz już wykonana przez entuzjastów.

Za jedyne 95 euro można zakupić zestaw do modyfikacji konsoli polegający na wymianie starego, oryginalnego ekranu na nowoczesny LCD. Po wymianie gry wyglądają zupełnie inaczej. Obraz, jak to na nowoczesnych ekranach - jest bardzo wyraźny. Dzięki temu stare tytuły na konsolę nabierają dosłownie nowych, lepszych kolorów. Dzięki nowszej technologii i niższemu poborowi prądu jeszcze dłużej możemy bawić się konsolą w pociągu czy na wycieczce bez podłączenia do sieci.

Atari Lynx modyfikacja LCD i wyjście VGA

Kolejną spektakularną zmianą jest wyjście VGA które pozwala na podłączenie do zewnętrznych odbiorników. Po podłączeniu do nowego TV można wygodnie grać, wyświetlać demo i na nowo bawić się naszym kociakiem...

Różnicę obrazu możecie zobaczyć na zdjęciach obok.

Piotr "Piter" Krużycki



Rampart na ekranie LCD.



Wyjście VGA z Atari Lynx.

Modyfikacja LCD i wyjście VGA

Cena: 95 euro (na razie tylko Lynx II)

Kontakt: odpowiedni temat na stronie:

atariage.com

lub napisz do nas - pomożemy!

Retro moda w pełni. Co mia-
sto to festiwal retro kompute-
rów. Idziemy na taki,
wspominamy z łezką w oczu.
Wracamy do domu, idziemy
do piwnicy, wygrzebujemy
stary karton z Amigą, Atari
czy Spectrum, odpakowujemy
i znów czujemy się jak dziec-
ko. Składamy wszystko do ku-
py, sięgamy kablem do
telewizora a tam... nie ma złą-
cza (sic!). Okazuje się że nasz
najnowszy TV 3D 8K Curved
Smart TV jest za głupi żeby
zrozumieć sygnał jaki generu-
je retro sprzęt.

Para opada, entuzjazm znika, pojawia
się smutek... Na szczęście to nie ko-
niec, ale nie sprzedajmy faktów.

Zmierzenie się z nowoczesnym telewi-
zorem nie jest aż takie trudne dla retro
sprzętu jak mogłoby się wydawać. No-
woczesne TV bardzo rzadko mają złą-
cza EURO-SCART, a S-Video wymarło
już kilka lat temu. Przy odrobinie szczę-
ścia może znajdziemy złącza Chinch.
Gniazdo RF (tradycyjne antenowe) cią-
gle jeszcze jest standardem, ale zazwy-
czaj jakość obrazu przez nie
przesyłanego jest odległa od zadowala-
jącej. Dlatego dziś najlepszym rozwią-
zaniem jest kupno konwertera RGB do
HDMI.

Sam zakupiłem taki dekodery kilka mie-
sięcy temu, prosto z Chin, poprzez je-
den z popularnych portali zakupowych.
Ponieważ producentów tego typu urzą-
dzeń jest wielu, a ich urządzenia są
z wyglądu identyczne, kierowałem się
kryterium ceny i możliwości, które są
zazwyczaj bardzo podobne. Różnice
ograniczają się do "przechodniego" por-

HDMI w służbie retro

tu HDMI oraz rodzaju generowanych
sygnałów. Patrząc na cenę, starałem
się nie wybierać możliwe najtańszej
oferty, uważając, że jest to kontrproduk-
tywne.

W ten sposób, po wydaniu około 150
PLN i odczekaniu 10 dni na paczkę,
stałem się właścicielem konwertera
RGB do HDMI. Całość robi naprawdę
dobre wrażenie – nie jest to plastikowa
taniocha, a elektronika zamknięta jest
w solidnej, metalowej obudowie.

Dekoder przetestowałem z dwoma
komputerami: Atari 65XE z VBXE i Atari
ST. W przypadku 65XE, nie generuje
on sam sygnału RGB – potrzebna jest
to tego odpowiednia przeróbka lub wła-
śnie VBXE. W przypadku Atari ST nie
potrzebujemy nic poza kablem.

Jakość obrazu jest lepiej
niż zadowalająca. Kolory
są pięknie nasycone, na-
wet programowe tryby
Interlace nie robiły na
dekoderze specjalnego
wrażenia. Jedyną ano-
malią jest skakanie obra-
zu o jedną linię podczas
animacji obiektów na
ekranie, ograniczone
jednak tylko do obszaru
tych obiektów. Jest to ar-

tefakt algorytmu usuwania przeplotu,
który nie do końca dawał sobie radę
z 65XE. W przypadku ST było to
znacznie mniej widoczne. Dodatkowym
problemem jest wprowadzanie przez
konwerter niewielkiego opóźnienia do
obrazu (i dźwięku) przesyłanego do te-
lewizora, co może utrudnić nieco granie
w gry akcji.

Zakupiony przeze mnie dekodery oce-
niam bardzo pozytywnie. Spodziewa-
łem się większych problemów,
szczególnie, że zakupu dokonywałem
nieco w ciemno. Mam nadzieję że chiń-
scy producenci z czasem wyeliminują
problem opóźnienia. Poza tą niedogod-
nością urządzenie mogę zdecydowanie
polecić każdemu.

Adam Kłobukowski



Rampart na oryginalnym wyświetlaczu.



Magiczne pudełko podłączone do telewizora.



FPS-y na mniejszego kota

Na Atari Lynx powstało kilka gier z gatunku FPS, postaram się opisać kilka z nich. Nie są to oczywiście wszystkie pozycje, ale te, w które udało mi się zagrać dłużej i mam wyrobioną opinię na ich temat.

Cybervirus

Gra wydana komercyjnie, nadal do kupienia w:

songbird-productions.com



wielu wrogów. Nasz bohater porusza się we wszystkich kierunkach, jak to przystało na FPS, a akcja jest bardzo szybka. Polecamy tę grę.

Luchsenstein 3D

Tak, to Wolf 3D na naszego Lynxa! Wspaniale robiony - ponieważ na razie mamy w pełni grywalną wersję demo - klasyk FPS z bardzo szybką grafiką



Rok produkcji: 2002

Ocena:

grafika:



dźwięk:



grywalność:



za 49 dolarów. Twoim zadaniem jest zatrzymanie hordy bio-mechanicznych wrogów, którzy pragną zaprowadzić swój porządek świata. Jesteś ostatnim z partyzantów, na którego liczą mieszkańcy Ziemi!

Gra jest perfekcyjnie zrobiona przez firmę Beyond Games - to ta sama, która na Jaguarze zrobiła świetnego Ultra Vortexa. Do wyboru mamy wiele rodzajów broni, m.in. laser, granaty oraz rakiety. Rozgrywka toczy się w 16 misjach oraz przy udziale

Rok produkcji: prace od 2013

Ocena:

grafika:



dźwięk:



grywalność:

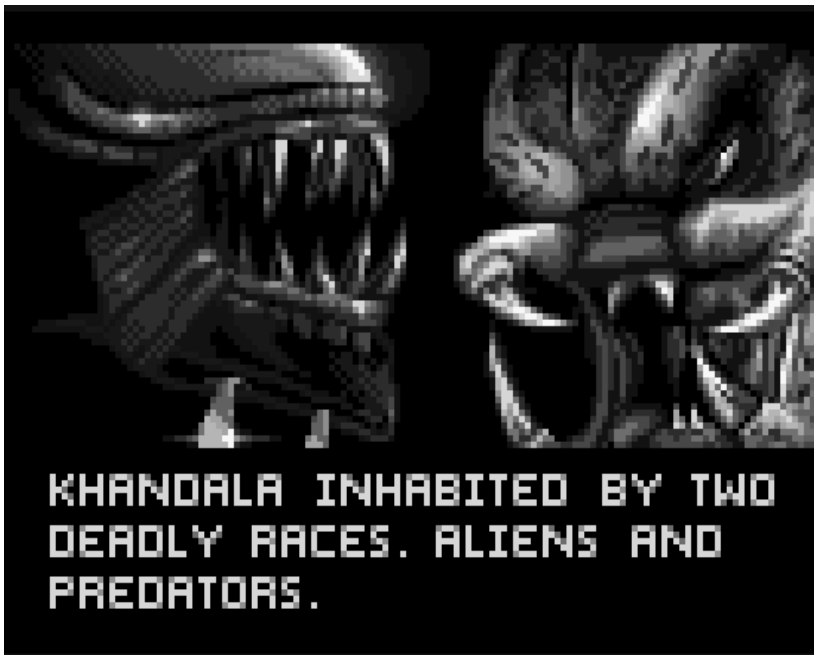


(20-30 fps), wieloma wrogami i wszystkim, co oferuje stary dobry tytuł!

Jest szansa, że gra po ukończeniu zostanie wydana na carcie. Na razie można w nią pograć mając FlashCarda lub na emulatorze.

Alien vs Predator

Grywalna wersja prototypu wielkiego hitu z Jaguara, który nigdy nie został skończony. Oczywiście można ściągnąć sobie ROM i pograć, ale gra jest wolniejsza od wspomnianego wcześniej Luchsenstein 3D, a także mamy możli-



Rok produkcji: 1991

Ocena:

grafika:



dźwięk:



grywalność:



pomysł to jej atuty, polecam ją fanatykom tego typu gier.

Turbo Sub

Jedna z pierwszych tego typu gier, ale czy to FPS to nie wiem do końca, bardzo szybka, dobra fabuła, dwie opcje do wyboru: walka na powierzchni wody lub pod wodą - zabijasz wszystko co zobaczysz, a więc typowa strzelanka.

Rok produkcji: 1993

Ocena:

grafika:



dźwięk:



grywalność:



wość ruchu w 4 kierunkach (podobnie jak Dungeon Master). Dostępne są trzy różne wersje, trzech różnych bohaterów, ciekawa grafika oraz obrazki pomiędzy przejściami. Można zabijać wrogów, zbierać przedmioty, uruchomić mapę, działa HUD... Szkoda, że gra nie została wydana, bo podobno skończona jest w 85%.

możliwością gry w maksymalnie czterech graczy w sieci poprzez ComLynx! bardzo szybka wektorowa grafika plus sprawdzony



Battlezone 2000

Nowa (jak na tamte czasy) wersja starego hitu automatuowego pod tym samym tytułem, znacznie poprawiona z nowymi power up'ami, nowymi wersjami broni, wrogami oraz

Rok produkcji: 1993

Ocena:

grafika:



dźwięk:



grywalność:





Nad dźwiękami mogliby więcej popracować. Mamy możliwość ruchu we wszystkich kierunkach, zwraca uwagę także bardzo szybkie scrollowanie. Pamiętajmy, że gra powstała w 1991 roku i jest dość łatwo dostępna na aukcjach.

Steel Talons

Bardzo dobry symulator lotu, widok z pierwszej perspektywy - zabawa jest naprawdę wyśmienita. Gra powstała jako automat arcade, mamy też wersję na Falcona. Producent to sama firma Atari. Mamy do dyspozycji śmigłowiec oraz różne misje do wykonania. W ukończeniu powierzonych zadań mamy do pomocy działko pokładowe, rakietę oraz mapę. Świetny klimat gry arcade! Mogłby być tylko szybszy.



Rok produkcji: 1991

Ocena:

grafika:



dźwięk:

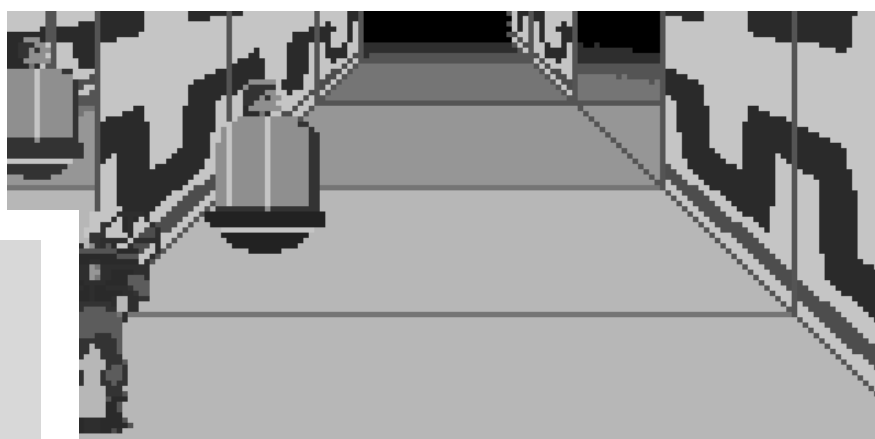


grywalność:



Xybots

Bardzo popularna gra na Lynxa, podobnie jak wcześniej omawiany tytuł, jest to świetny port z wersji arcade! Dźwięki w grze, jak i muzyka w tle są bardzo dobre. Razem tworzą świetną atmosferę w przechodzeniu kolejnych labiryntów, gdy zabijamy kolejnych wrogów. Jeszcze lepiej można się zabawić gra-



Rok produkcji: 1991

Ocena:

grafika:



dźwięk:



grywalność:



jąc jednocześnie w dwóch grach - także poprzez port ComLynx! Każdy ma swojego Lynxa, a na ekranie jesteście razem w tym samym momencie. Być może zostanie zrobiony nawet Doom na Lynxa. Prób jeszcze nie było, ale konsola

cały czas żyje, nowe gry wychodzą w miarę regularnie, choć może nie tak często jak na jego większego kociaka - Jaguara, ale o tym opowiem w następnym Retrokompie. Do zobaczenia!

Piotr "Piter" Krużycki

Polskie znaki w systemach TOS, MiNT i Magic

Do napisania tego artykułu nakłonił mnie mój serdeczny kolega Piotr "Piter" Krużycki. Problem instalacji polskich znaków na komputery 16/32 bity objawił się natychmiast wraz z pojawieniem po raz pierwszy komputerów serii ST, czyli ok. 1985 roku. W Internecie można znaleźć wiele rozwiązań - nakładek na system w postaci akcesoriów, np. "Mikołaj Rej". Ja używam już od wielu lat na wszystkich moich komputerach, czyli Atari Falcon 030/060, Atari TT oraz Hades, programu "Polskie Znaki" w wersji 3.02 autorstwa Michała Kowalskiego.

W artykule tym chciałbym dokładnie opisać instalację polskich znaków w systemie TOS, MiNT oraz Magic, a także używania ich w podstawowych aplikacjach, czyli w edytorach tekstów takich jak QED korzystających z systemowych czcionek, jak Papyrusa korzystającego z czcionek wektorowych *.SPD oraz windowsowych *.TTF, a na koniec opisać instalację dla flagowego programu do składu tekstu (Desktop Publishing - DTP), czyli Calamusa, który przede wszystkim korzysta z własnych czcionek wektorowych *.CFN.

Rozwiązanie problemu instalacji polskich znaków zacznę od systemu TOS. Do katalogu AUTO kopiujemy plik:

POL_FONT.PRГ

oraz do katalogu, w którym trzymamy pliki *.CPX panelu kontrolnego (XCONTROL.ACC, ZCONTROL.ACC lub COPS.ACC), kopiujemy odpowiedni plik:

POL_FONT.CPX

Najczęściej używanym katalogiem jest tu:

C:\CPX\

W pliku CPX ustawiamy w jaki sposób chcemy uzyskać polskie znaki - z odpowiednią kombinacją klawiszy ALT, CONTROL lub klawisz deadkey w zakładce "Ustawienia" (patrz zrzut ekranu). Ponadto w zakładce "Klawiatura" ustawiamy odpowiednio litery:

- Ą z klawiszem A
- ż z klawiszem Z
- Ź z klawiszem X

itd.

Ostatnią rzeczą jaką robimy to ustawienie w zakładce "Standard" kodowania polskich znaków - w trybie szesnastkowym. W moim przypadku używam do tego kodowania ISO 8859-2, którego szczegóły można znaleźć w Internecie na stronie:

https://pl.wikipedia.org/wiki/Kodowanie_polskich_znakow



Dokładnie się przyglądając widzimy, że w przypadku kodowania ISO-8859-2 polską literę "ą" uzyskujemy dziesiętnie jako 161 co odpowiada \$A1 w trybie szesnastkowym i analogicznie dla pozostałych polskich znaków. Na tym kończą się odpowiednie ustawienia w programie "Polskie znaki".

Do pełni szczęścia pozostaje nam odpowiednie zainstalowanie fontów zawierających polskie znaki. Nieocenionym programem jest nakładka NVDI od wersji 2.5 wzwyż. Po zainstalowaniu tego programu pozostaje odpowiednie wyedytowanie pliku:

ASSIGN.SYS

w którym wpisujemy odpowiednie czcionki dla różnych rozdzielczości ekranu i wielkości fontów, np.

```
01p SCREEN.SYS
s SYSTEM8.FNT
s SYSTEM9.FNT
s SYSTEM10.FNT
s MONACO20.FNT
```

Oczywiście wyżej wspomniane fonty, czyli np. SYSTEM10.FNT, muszą zawierać polskie czcionki w standardzie ISO, ponadto muszą znajdować się w katalogu:





C:\GEMSYS\

Myślę, że można je znaleźć bez większych problemów w Internecie, ewentualnie samemu przygotować przy wykorzystaniu dowolnego edytora fontów bitmapowych np. FONTKIT4.PRG. Ja takie fonty nabyłem razem z zakupem programu do polskich znaków "Polskie znaki" w starszej wersji 1.21, oczywiście na dyskietce. Miałem już odpowiednio przygotowane czcionki w standardzie ISO-8859-2 i nie tylko, choćby w standardzie ATARI-Calamus potrzebnym przy prawidłowym instalowaniu polskich znaków w programie Calamus. Jednak dokładny opis znajduje się w dalszej części tego artykułu.

Takie skonfigurowanie polskich znaków będzie umożliwiało pisanie polskich tekstów przy wykorzystaniu czcionek systemowych załadowanych w pliku:

ASSIGN.SYS

czyli pisanie przy wykorzystaniu edytorów tekstu korzystających z takich czcionek, czyli np. QED. Inaczej jest przy wykorzystaniu czcionek wektorowych czyli *.SPD albo *.TTF ładowanych do systemu przy pomocy programu NVDI. Czcionki te są stosowane w edytorach tekstu jak np. Papyrus.

W tym wypadku musisz mieć odpowiednio skonfigurowaną mapę czcionek. Jest to plik:

SPDCHAR.MAP

Trzeba go odpowiednio skonfigurować, a można to zrobić w programie CHAR-MAP5.APP, za pomocą którego możesz w tym samym pliku (czyli SPDCHAR.MAP) ustawić polskie znaki dla czcionek wektorowych TRUETYPE (windowsowych *.TTF). Tak zrobioną mapę można wykorzystać w Papyrusie i korzystać zarówno z czcionek *.SPD,

jak i czcionek *.TTF. Przy jakichkolwiek kłopotach proszę o kontakt. Wyżej wspomniany program można bez najmniejszych kłopotów odnaleźć na wielu serwerach w sieci.

Teraz przejdę do odpowiedniej konfiguracji czcionek, aby móc pisać po polsku w programie Calamus. Dla każdego użytkownika jest rzeczą naturalną, że korzysta on z własnych czcionek wektorowych *.CFN, a jest ich ogromna ilość. W tym wypadku sprawa jest o wiele prostsza i polega

tylko na wybraniu prawidłowych opcji w nakładce "Polskie znaki". W odpowiednim pliku:

CPX POL_FONT.CPX

w zakładce "Ustawienie" wybieramy:

Atari
oraz
polskie

i tak zapisujemy ustawienia.

Można też zmienić nazwę pliku PRG i CPX na inną np. POLFONT lub dowolną nazwę według własnego uznania, co pozwoli - przy odpowiednio skonfigurowanym programie podczas bootowania komputera, np. Xboot - wybierać konfigurację w zależności od potrzeb. Należy pamiętać, że jeżeli chce się także korzystać z polskich czcionek systemo-

wych, należy zmienić ładowanie czcionek w pliku:

ASSIGN.SYS

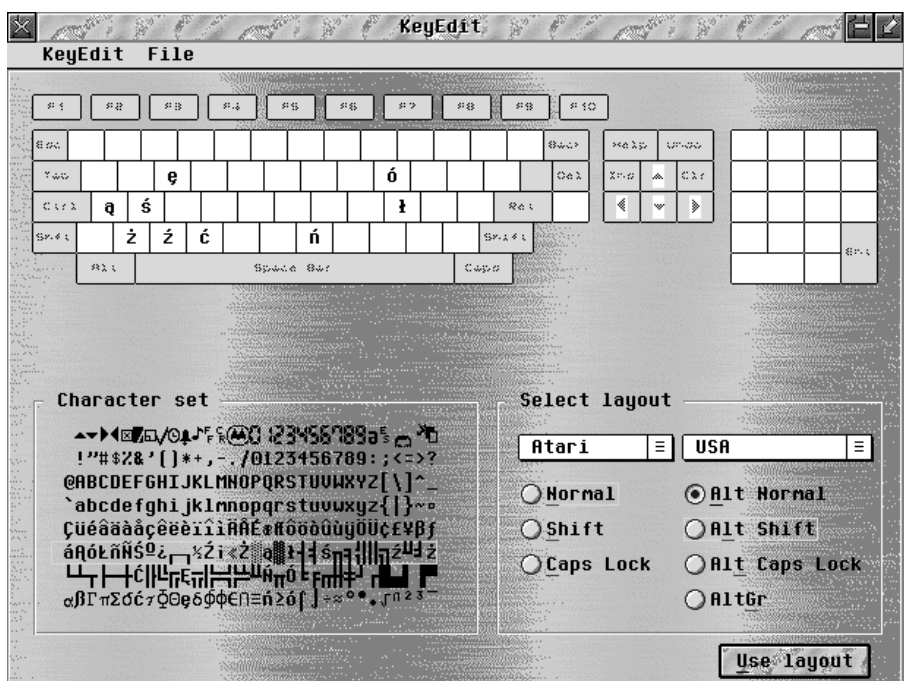
w którym podaje się czcionki mające ustawione polskie znaki przy wykorzystaniu kodowania Atari-Calamus. Ja u siebie w Xboot wybieram także odpowiedni ASSIGN.SYS.

Na tym kończy się pierwszy etap konfiguracji polskich znaków w systemie TOS. Dalej opiszę konfigurację polskich znaków w obu standardach przy wyko-



rzystaniu innych popularnych systemów, czyli MiNT oraz Magic.

W przypadku MiNTa jest to łatwiejsze, a sama konfiguracja jest bardzo uproszczona - wraz ze sposobem przełączania pomiędzy zainstalowanymi, a więc dostępnymi kodowaniami polskich znaków.





W systemie MiNT, od FreeMiNTa w wersji 1.16.1, odpowiednia konfiguracja wymaga ustawień w pliku:

XAAES.CNF

O ile oczywiście korzystamy z tej zakładki na system. W pliku znajdujemy linię:

"Switch keyboard tables"

w której wpisujemy nazwy zainstalowanych klawiatur:

```
keyboards = <, calamus,  
british
```

Jak dokładnie należy to zrobić jest opisane w samym pliku konfiguracyjnym XAAES.CNF, w części dotyczącej przełączania pomiędzy klawiaturami:

CALAMUS.TBL

oraz

BRITISH.TBL

przy pomocy kombinacji klawiszy, np. ALT+CTRL+<. Wspomniane pliki muszą znajdować się w katalogu, czyli przykładowo:

C:\MINT\1-19-CUR

Ponadto należy pamiętać oczywiście o czcionkach *.FNT, podobnie jak w przypadku TOSa, ustawionych w pliku ASSIGN.SYS.

Do edycji klawiatur służy program dołączony do dystrybucji FreeMiNT o nazwie KEYEDIT.APP, a wszystko odbywa się w prosty i przejrzysty sposób. W polu "Character set" pokazuje nam się zainstalowana czcionka systemowa. W polu "Select layout", tak jak to pokazuje zrzut ekranu, wybieramy:

Atari

oraz klawiaturę, jaką mamy podpiętą do komputera, czyli UK, USA, FRANCE itp. Następnie wybierając małe i duże litery ALT, ALT+SHIFT ustawiamy poprzez odpowiednie wybieranie z pola "Character.set". W ten sposób przygotowujemy sobie alternatywne kodowanie np. dla programu CALAMUS.TBL.

Na zakończenie tego artykułu pozostaje tylko opisanie konfiguracji polskich znaków w przypadku systemu Magic. Tutaj, jeśli mamy już poprawnie ustawione polskie znaki pod czystym TOSem, sprawa jest banalnie prosta. Konfiguracja odbywa się w identyczny sposób. W zależności od wybranego standardu kodowania należy uruchomić program

w katalogu AUTO oraz CPX, czyli odpowiednio:

POL_FONT.PRGS

oraz

POL_FONT.CPX

pamiętając o załadowaniu odpowiedniego pliku ASSIGN.SYS. W ten oto sposób uzyskujemy polskie znaki zarówno w systemie, jak i w aplikacjach korzystających z czcionek wektorowych np. Papyrus.

Podsumowując, mam nadzieję, że poniższy tekst przyda się wielu użytkownikom komputerów 16- i 32-bitowych, począwszy od 520 ST, a skończywszy na klonach MILAN, HADES czy FIREBEE. Wszelkie uwagi o niedoskonałościach artykułu proszę kierować do autora. W razie niejasności postaram się w rzetelny sposób rozwiązać każdy problem.

W niedalekiej przyszłości opiszę dokładną konfigurację przy wykorzystaniu czcionek wektorowych *.SPD oraz *.TTF, czyli sposób odpowiedniego przygotowania mapy pliku SPD-CHAR.MAP.

Piotr "Kroll" Mietniowski

Zapraszamy do współpracy przy tworzeniu magazynów:

RETROKOMP oraz **AMIGAZYN**

Masz ciekawy sprzęt retro?

Napisalesz program, grę lub demo?

Kolekcjonujesz gry na dyskietkach?

A może masz pomysł na nowy artykuł?

NAPISZ DO NAS: a2@amiga.net.pl



CALAMUS 2015

Calamus, legendarny program DTP dostępny na nasze Atari, a także na Windowsa oraz Mac OS X, od prawie dziewięciu lat nie był oficjalnie upgradowany. Nie jest to do końca prawdą, ponieważ przez te lata mieliśmy około siedmiu bezpłatnych aktualizacji, a także nowych dodatków dla zarejestrowanych użytkowników Calamusa SL w wersji 2006. Przez te lata dużo się zmieniło, wiele osób zgłaszało potrzebę większego wsparcia dla popularnych formatów, takich jak PDF, EPS, RTF i WMF, w tym zaimplementowania ich do standardowej wersji programu.

W 2015 roku firma Invers Software wydała oficjalną, nową wersję programu, Calamus SL 2015. Poprawionych zostało wiele szczegółów typu przyciski, zostały uproszczone opcje, m.in. ścieżki dostępu, pojawiły się nowe opcje typu info, poprawiono wiele funkcji jak na przykład ulepszone kopiowanie i przemieszczanie obiektów z ramek do schowka oraz w drugą stronę. Zwiększona została też stabilność programu.

Jednak najważniejszymi, zauważalnymi zmianami stało się dołączenie bardzo wielu modułów, które do tej pory były opcjonalne i dodatkowo płatne do wersji standardowej. Autorzy musieli to zrobić, ponieważ w dzisiejszych czasach opcja importowania plików w formacie PDF, PS, WMF i innych to podstawa! Do tej pory były one dostępne w wersji pełnej lub była możliwość zakupu osobno, jako uzupełnienie wersji standardowej.

Oczywiście wszystkie fonty oraz dokumenty ze starszych wersji są poprawnie otwierane przez najnowsze wydanie

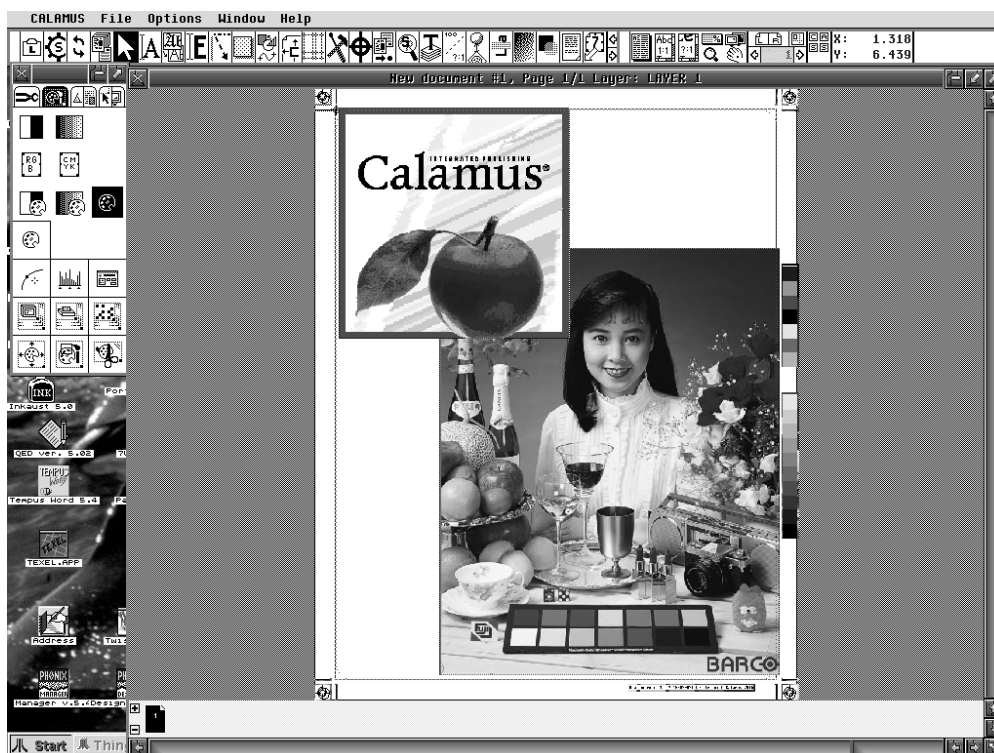
Nowe podejście?

programu. Przez cały 2015 rok doczekaliśmy się aż trzech większych lub mniejszych aktualizacji, co może oznaczać dalszy rozwój tego wspaniałego narzędzia dla DTPowców. Poprawiła się moim zdaniem też jakość kontaktów z autorami Calamusa w sprawach problematycznych lub w związku z ładowaniem modułów.

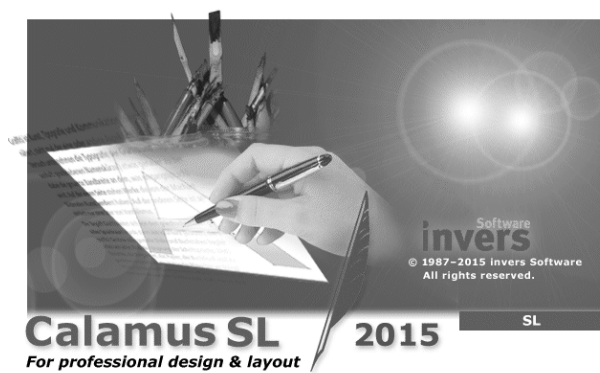
Z kolegą Piotrem Mietniowskim nabyliśmy pełną wersję programu, wraz ze wszystkimi modułami, które wcześniej nie były dla nas dostępne. Pozwoliły na to ogłoszone promocje. I tak na przykład, można było nabyć program w cenie niższej nawet o 50% i dodatkowo upgradując go ze starej wersji sprzed kilkunastu nawet lat, bez kupowania pełnej wersji.

Ciekawym modulem jest moduł pozwalający podłączyć do naszego komputera ploter tnący i wycinać na nim różnego rodzaju grafiki z folii samoprzylepnej. Calamus SL 2015 jak zwykle oferuje 100% WYSIWYG z technologią SoftRippingu, idealne odwzorowanie rastra przy wyjściu na urządzenia zewnętrzne, zoom do 999,999% (sic!), precyzję do 1/10'000 mm, łatwe przygotowanie tzw. Master Page, idealną justyfikację tekstu na każdej, nawet najbardziej skomplikowanej ścieżce wektorowej, nielimitowaną ilość stron dokumentu i wiele, wiele innych opcji.

Program Calamus SL 2015 można uruchomić na komputerze Atari w rozdzielczości minimum 800x600, tak więc



Calamus w trakcie edycji dokumentu.



musimy mieć przynajmniej Atari TT030 z monitorem Atari TTM194/195 lub kompatybilnym innej firmy np. Protar, a także z monitorem LCD poprzez nowy produkt z USA - adapterem Tenox, który pozwala na uzyskanie wysokiej rozdzielczości tzw. TT-High (czyli 1280x960 - patrz opis w tym numerze Retrokomp) lub Atari TT030 z kartą graficzną - patrz informacja o nowych adapterach do kart graficznych zamieszczona obok.

Idealnym sprzętem są szybkie klony Atari takie jak Medusa lub Hades. Są one jednak trudno dostępne i zazwyczaj osiągają wysokie ceny na aukcjach internetowych. Nowym klonem Atari jest Firebee, którego kolejna produkcja właśnie ruszyła, ale nadal trwają prace nad poprawieniem Calamusa tak, aby działał poprawnie na nowej maszynie. Jesteśmy w ciągłym kontakcie z autorami programu i mamy nadzieję, że po tym, jak już nowe Atari trafi na nasze biurko, problem ten zostanie rozwiązany.

Calamusa można także uruchomić na każdym - zarówno nowym, jak i starszym systemie Windows. Nie jest to jednak typowy "okienkowy" program, ponieważ przy starcie inicjuje się emulator software'owy TOS, a następnie sam Calamus. My tego właściwie nie zauważamy, po uruchomieniu mamy takiego samego Calamusa jak na Atari. Dzięki temu mamy możliwość pracować w środowisku dobrze nam znanym i nie musimy uczyć się nowego wyglądu programu.

Trzecim systemem na którym uruchamiamy Calamusa jest Mac OS X. Aby to zrobić musimy jednak zakupić dodatkowo komercyjną wersję programu MagicMac X, który tworzy środowisko kompatybilne z TOS - znanego Magica - na którym podobnie jak na na Windowsie uruchamiamy Calamusa SL 2015. Wadą jest, że w tym momencie MagicMaca X możemy uruchomić na Mac OS X 10.6.8 lub na starszej wersji,

bo nie wspierane są nowsze systemy. Plussem jest natomiast fakt, że MagicMac X to pełen system i większość TOSowych programów można na nim uruchomić - w pełnym kolorze i w rozdzielczości jaką oferuje nam Mac.

Możemy więc pracować m.in. na Texelu - świetnym arkuszu kalkulacyjnym, pakiecie

office - Papyrusie, grach, które działają pod czystym TOSem, ponadto wielu programach FTP, pocztowych i innych.

W następnym numerze Retrokompa zaczniemy robić projekty oraz pokażemy sztuczki jakie można wykonać w tym świetnym programie, podzielimy się naszym 20-to letnim doświadczeniem w pracy z tym narzędziem DTP. Będziemy informować o nowych promocjach, a być może uda nam się nawet uzyskać specjalne promocje dla Czytelników naszego pisma! Przeczytajcie też artykuł o polskich znakach napisany przez Piotra Mietniowskiego, który pomoże Wam uzyskać „ogonki” w Calamusicie. Jeśli macie jakieś pytania - piszcie do mnie! Do zobaczenia w następnym numerze Retrokomp.

Piotr "Piter" Krużycki
piotr@kni.com.pl

Adaptery kart graficznych do Mega ST i TT

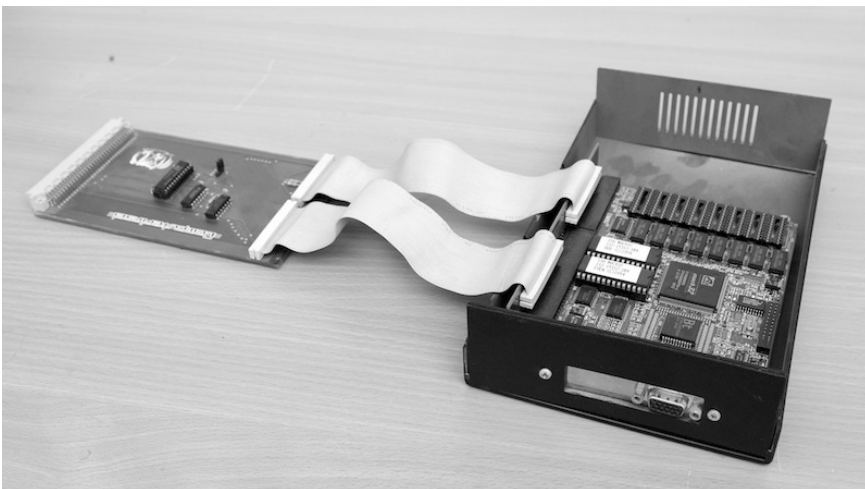
Nasz dzielny kolega Articwroc z forum dyskusyjnego na stronie www.atari.pl zaczął przeprojektowywać stare rozwiązania adapterów kart graficznych, które można podłączyć do komputerów Atari Mega ST i TT, a także Mega STe.

Są to dwa rozwiązania - adapter karty graficznej NOVA oraz adapter karty graficznej ISA ET4000. Polecam

zainteresować się osobom, które pragną zacząć projektować w nowym Calamusicie lub chcą sobie usprawnić pracę instalując kartę graficzną do standardowej konfiguracji TT-ki. Więcej informacji znajdziesz na forum:

www.atari.pl

Cena: nieznana
Kontakt: forum dyskusyjne





TenoxVGA adapter do TT-ki

TenoxVGA to adapter do komputera Atari TT030, wyprodukowany przez naszego rodaka Antoniego Sawickiego, który mieszka w USA. Urządzenie pozwala na uzyskanie na nowoczesnych monitorach LCD lub starszych CRT, wysokiej rozdzielczości - tak zwanej TT-High tj. 1280x960.

Wcześniej taką rozdzielczość można było uzyskać tylko na specjalnych monitorach ECL produkowanych przez Atari - modele TTM194, TTM195 lub innych firm, na przykład Protar czy Eizo, ale i tak to były specjalnie produkowane monitory do podłączenia naszej TT-ki,

a co za tym idzie ciężkie do zdobycia, jak i ciężkie w dosłownym tego słowa znaczeniu - ważą one ok. 20-30 kg. Adapter wykonany przez pana Antoniego podłączamy do standardowego złącza monitorowego w TT030, a następnie podłączamy się do, na przykład, nowoczesnego monitora LCD 19" i uzyskujemy bardzo dobrej jakości obraz, dokładnie taki jak we wcześniej wspomnianych monitorach pracujących w systemie ECL.

Aby jednak obraz był tak idealny należy poszukać odpowiedniego monitora. Wszystkie właściwie monitory LCD potra-

fią obsłużyć rozdzielczość 1280x1024. Jeśli monitor nie ma odwzorowania do 1 pixela (ta opcja nazywana jest najczęściej expansion off), mogą wystąpić zakłócenia, które uniemożliwią wygodną pracę. W moim doświadczeniu najlepiej nadaje się do tego monitor NEC 1990SXi - trzeba zwrócić uwagę na cały symbol. Podobno inne modele też działają poprawnie, ale model Nec 1990 FXp nie nadawał się. Jeśli ktoś zdecyduje się i kupi adapter, najlepiej wypróbować go na monitorach, które akurat posiadamy, a jeśli nie to szukać podanego przeze mnie modelu, tym bardziej, że jest idealnie pasujący kolorem i zgrany swoją stylistyką z naszą ukochaną TT-ką. Podejrzewam, że adapter poprawnie będzie działał także z Atari Mega ST wyposażonym w kartę graficzną Protar 4ML lub Atari Viking, a także Atari Mega STe z kartą E-Screen. Są to karty pozwalające właśnie na podłączanie monitorów Atari TTM-194/195.

Adapter jest świetnym urządzeniem, praca na TT-ce i Cubase, gdzie mamy do dyspozycji dużo więcej ścieżek lub na Calamusie, gdzie możemy obejrzeć stronę A4 w skali 1:1 na monitorze staje się nową jakością.

Piotr "Piter" Krużycki
piotr@kni.com.pl



fią obsłużyć rozdzielczość 1280x1024. Jeśli monitor nie ma odwzorowania do 1 pixela (ta opcja nazywana jest najczęściej expansion off), mogą wystąpić zakłócenia, które uniemożliwią wygodną pracę. W moim doświadczeniu najlepiej nadaje się do tego monitor NEC

Rok produkcji: 2013
Producent: Antoni Sawicki

Platforma: Atari TT030, Mega ST z kartą graficzną Protar 4ML lub Atari Viking, Mega STe z kartą E-Screen

Cena: od 100 dolarów
Dystrybucja: www.tenox.net



uIP

łatwiej się nie da

uIP to fantastyczne narzędzie do szybkiego i łatwego przetrzucania plików na nasze duże Atari.

Nie potrzebujemy do tego żadnych sterowników, musimy mieć jedynie adapter NetUSBee. W tym celu zapraszam na stronę:

www.lotharek.pl

Adapter należy podłączyć do routera i rzecz jasna do Atari. Dalej uruchamiamy program uIP, który po krótkiej chwili automatycznie uzyskuje adres IP wyświetlając go na ekranie naszego monitora. Po drugiej stronie uruchamiamy komputer klasy PC lub Mac z przeglądarką internetową, gdzie wpisujemy uzyskany przez Atari adres IP i otwiera nam się okno naszego dysku oraz stacji dyskiety, a więc mamy pełen dostęp do zasobów naszego Atari.

Teraz już wystarczy tylko przetrzucać pliki na Atari i cieszyć się nowym oprogramowaniem. Program napisany jest w języku C przez Sqwarda - znanego programistę scenowego i jest cały czas rozwijany. Na niektórych TT-kach zauważono problem z uzyskaniem adresu IP przez program, niestety przyczyna na razie nie znana. Dlatego z niecierpliwością czekamy na kolejne wersje.

Piotr "Piter" Krużycki

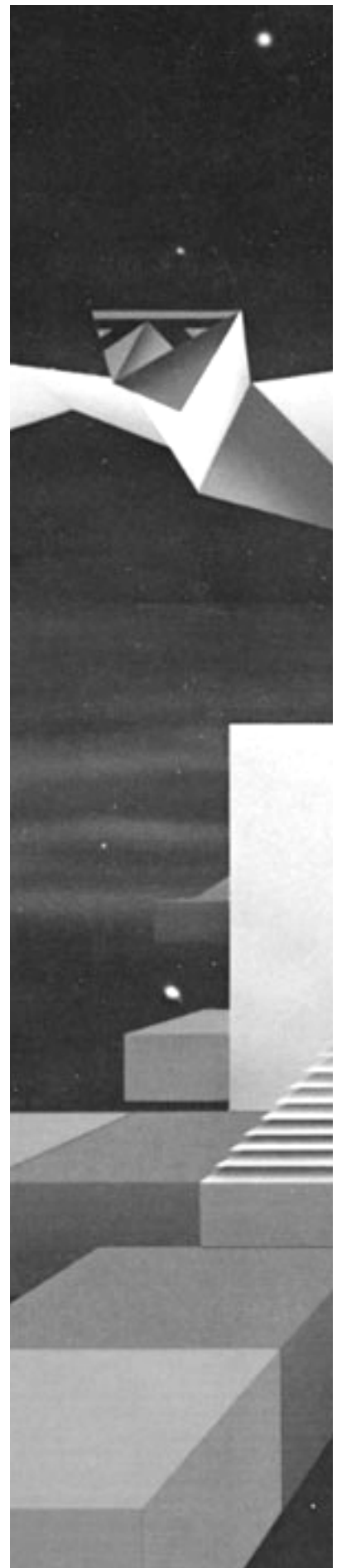
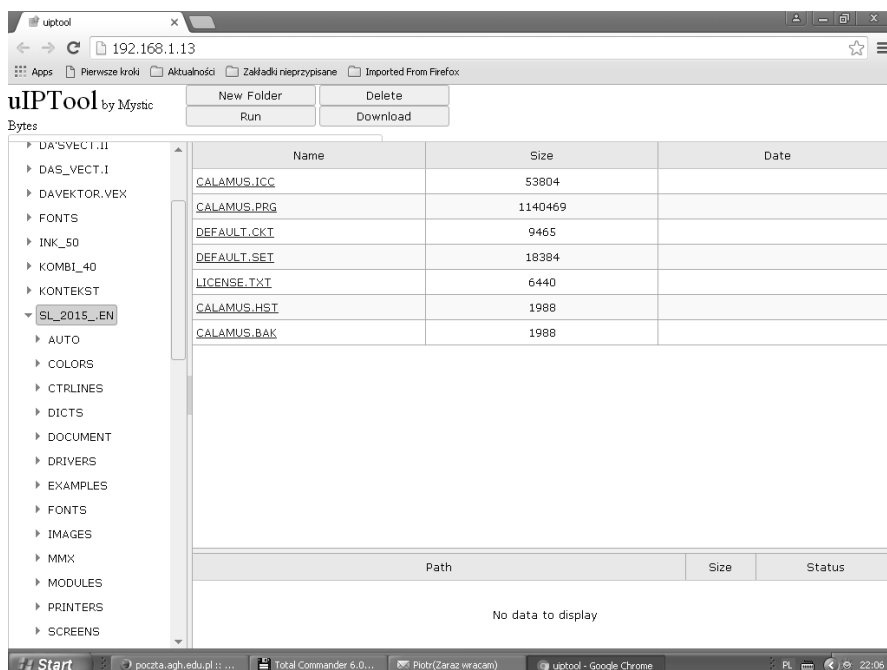
Ostatnia wersja: .43 (2015.12.07)

Autor: Mariusz Buras 'Sqward'

Status: freeware ze strony:

bitbucket.org/sqward

Platforma: każde Atari ST/STe, TT030 i Falcon 030/060, Firebee, Hades i inne klony





WYKOPALISKA



W pierwszym numerze Retrokompas chciałbym zapoczątkować dział, w którym będę prezentował gry napisane tylko i wyłącznie na Atari i to zaczynając od konsoli VCS, a na Falconie oraz Jaguarze kończąc. Będą to tytuły albo wyjątkowo dobre, albo bardzo pomysłowe lub zupełnie nietypowe, czyli coś, co pozwoli zainteresować się nimi nie tylko graczom z Atari, ale także fanom innych platform. Wszak ostatnio wielu użytkowników Amigi kupuje STe, a większość chciałaby mieć u siebie Falcona. Podobnie dzieje się też w drugą stronę. Dożyliśmy czasów, gdzie często widzi się A500 i ST na jednym biurku.

hammerze mamy mroczną, cyberpunkową atmosferę z niesamowitą muzyką i klimatycznymi dźwiękami. W grze jesteśmy pilotem tytułowego Skyhammera, zostaliśmy zatrudnieni przez firmę Cytos do ochrony miasta oraz pilnowania nieruchomości i interesów pracodawcy. Gra jest świetną strzelanką 3D, ze swobodnym ruchem we wszystkich kierunkach. Do dyspozycji mamy albo myśliwiec, albo śmigłowiec oraz całą masę różnych rodzajów broni od lasera po rakietę, zestawy naprawcze naszego statku, mapę z GPS i inne. Misje, które mamy wykonać są



W grze latamy nad trzema różnymi miastami, każde z nich jest inne, ma inny układ ulic, budynków, a więc przed nami różnorodne wyzwania.

Gra Skyhammer jest podobna do gier z innych platform i same nasuwają się dwa tytuły: G-Police oraz Descent. Polecam ją każdemu fanowi tego typu tytułów, tym bardziej, że nadal można ją kupić nową, w 2015 roku była reedycja.



Dziś chciałbym opisać dwie gry: Skyhammer na Jaguara - za jego klimat, a także Skystrike Plus za zabawę jakiej dostarcza. Skyhammer została napisana przez grupę Rebellion - tę samą, która zrobiła Alien vs Predator na naszego Jaguara, co od razu stawia ją w gronie potencjalnie świetnych tytułów. W Sky-

nieliniowe i ciekawe. Do dyspozycji są dwie opcje gry: Mission Mode i Battle Mode. W pierwszej wykonujemy zadane misje i powiększamy terytorium wpływów naszej korporacji, w drugiej natomiast możemy zmierzyć się z niekończącą się hordą wrogów.

SKYHAMMER

Platforma: Atari Jaguar

Nośnik: cart

Autor: Rebellion (1998)

Wydanie: Songbird Productions (2000)

Dystrybucja: songbird-productions.com

Cena: około 90 dolarów

Ocena:

grafika:

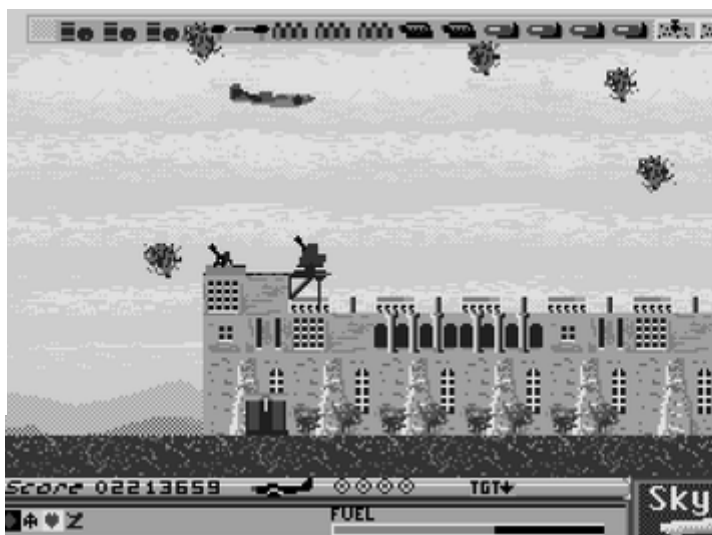


dźwięk:



grywalność:





SKYSTRIKE PLUS

Platforma: Atari ST

Nośnik:

Autor: Shadow Software (1990)

Ocena:

grafika:



dźwięk:



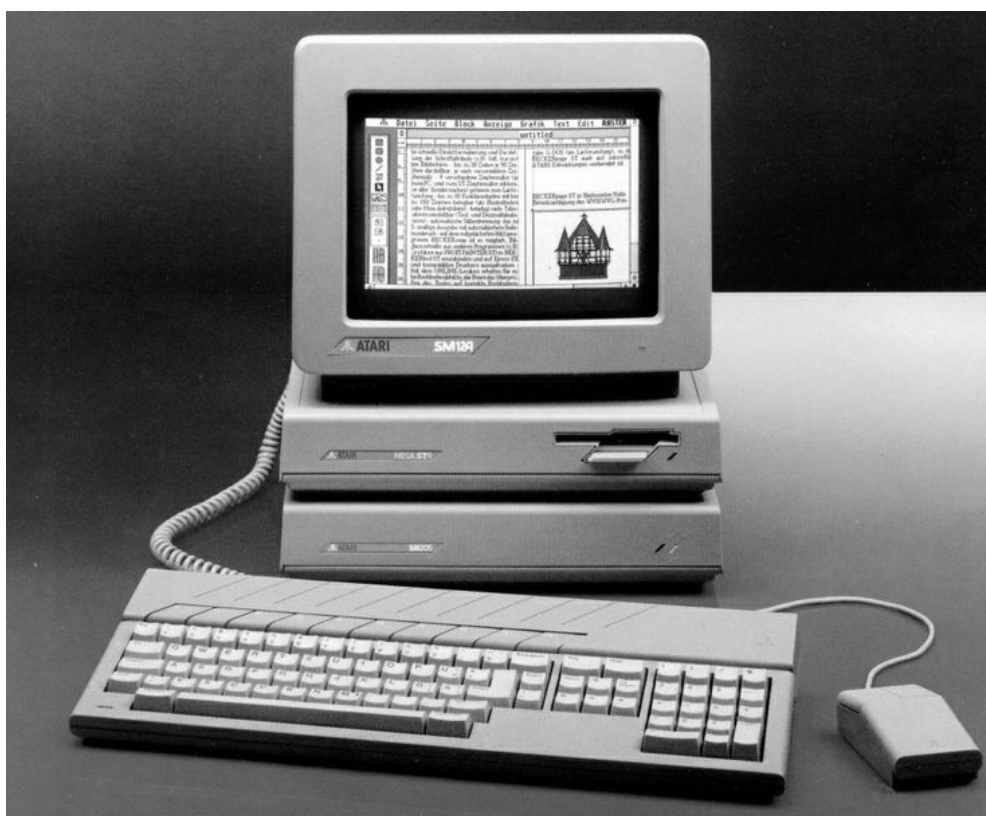
grywalność:



SkyStrike Plus to dobra strzelanka 2D, gdzie sterujemy samolotem i próbujemy zlikwidować naszych wrogów. Mamy do dyspozycji 56 misji, różne lokacje, dobrze zrobiona jest także fizyka lotu, swobodne opadanie, lądowanie i tym podobne akrobacje w powietrzu. Uszkodzenia naszego Spitfire'a są różnorodne, od przestrzelenia baku i szybkiej przy tym utracie paliwa, uszkodzenia hamulców (będziemy wtedy mieć problemy przy lądowaniu) do uszkodzenia silnika, które objawia się tym, że za na-

szym samolotem pojawia się chmura czarnego dymu. Mamy przeciw sobie wielu wrogów: inne samoloty, czołgi, działa przeciwlotnicze, możemy też niszczyć cele ziemne jak fabryki i lotniska. Gra została napisana w języku programowania STOS, wykonana jest dobrze, poziom realizmu jest wysoki, dzięki temu SkyStrike może przyciągnąć nas na dłuższy czas do ekranu Atari ST. Grę można łatwo znaleźć w Internecie, jeśli będziesz miał z tym kłopot - napisz do mnie.

Piotr "Piter" Krużycki



ATARI MEGA ST4

**CZYTAJ
W
RETROKOMPIE!**



Hot Style

dyskietka do poduszki

Magazyny dyskowe były jednym z nieodłącznych elementów sceny komputerowej ostatniej dekady XX wieku. Pojawiały się na wszystkich popularnych platformach, stanowiąc podstawowe źródło wiedzy na temat nowych produkcji, ciekawostek technicznych, obyczajów, jak i spotkań komputerowych zapaleńców. Czy w dobie powszechnego dostępu do Internetu wydawanie magazynów na dyskietkach będzie przedsięwzięciem spisanim na straty? Zdecydowanie nie, czego dowodzą dwie polskie formacje: Tropyx i Draco.

Wspólnymi siłami obie grupy od 2014 roku wydają periodyk o tytule „Hot Style”. Ów discmag traktuje nie tylko o wiekowych komputerach (głównie Commodore 64), ale sięga również po tematykę poza scenową. Trzonem zawartości tematycznej są różnorodne publikacje około komputerowe: recenzje gier, relacje z imprez (partiesy, wystawy, zloty graczy), opisy sprzętowe i nowości ze sceny komputerowej C64. Liczne grono autorów publikacji tworzy artykuły na temat swojego hobby, jak i wspomnień z odległej komputerowej przeszłości.



Podjęmowane są na łamach Hot Style również tematy bieżące, takie jak chociażby technika wydruku 3D i animacja. „Hot Style” to także miejsce dla twórców dzieł literackich.

Obok opowiadań i wierszy pojawiły się w trzecim numerze magazynu limeriki, a poprzednie wydanie zawiera humorystyczną scenkę teatralną, inspirowaną dokonaniem nieistniejącego już kabaretu Potem. Z kolei w „dziewiczym” wydaniu magazynu przeczytać można humorystyczną scenkę teatralną pt. „Wielka scena”. Poszczególne publikacje znajdują się w konkretnych działach,

a całość oprawiona jest przyjemną szatą graficzną i stonowaną ścieżką dźwiękową.

Redaktorem, koordynatorem, pomysłodawcą współtworzenia magazynu, korektorem i motywatorem jest V-12 z grupy Tropyx. Za kod magazynu odpowiada Hank/Draco. Od drugiego numeru magazyn wzbogacony jest komiksem autorstwa Sekatora. Jest on także twórcą wszystkich dotychczasowych coverów (okładek) na dyskietki. Do listy stałych współpracowników należą następujące osoby: Will, KaiN, Ryan, Sachy, V0yager, Johny, Factor6, SuperNoise, Slake, Archibald, Kyno i Marek Pelc. Oprócz nich autorami publikacji są: Arek, Rocky, Diony, Szczytu, CJ Warlock, Rav.en, Tom Rain, Gohanks, Dorota Kuczyńska i Simon Szarafin. Twórcy magazynu liczą, że lista ta z biegiem czasu będzie się powiększać. Obok wersji dyskietkowej Hot Style dostępny jest w wersji on-line pod adresem:

www.hotstyle64.com

Część tekstów wzbogacono zdjęciami. Na stronie udostępniono także obrazy dyskietek wszystkich magazynów oraz covery do wydrukowania. Magazyn ukazuje się dwa razy w ciągu roku. Zapraszamy do współpracy!

Paweł "V-12" Ruczek



1581

Na stację dyskó 1581 prędzej czy później natknie się każdy fan komputerów Commodore. O ile jej przydatność w roku Pańskim 2016 stoi pod sporym znakiem zapytania, (ze względu na SD2IEC, 1541 Ultimate i podobne wynalazki) nie oznacza to, że nie można nadal skorzystać z pocztowych dyskietek 3,5 cala.

Stacja, która posiadam została zakupiona jeszcze w kwietniu 2011 na aukcji internetowej. Urządzenie miało rozerwaną obudowę (widać gdzieś gdzie jeszcze nie znają śrubokrętów) i rozbebeszony mechanizm. Sprzedawane było w stanie nieznanym, ale nie przestraszyłem się tego faktu, gdyż posiadałem już wcześniej komplet scalaków do 1581 (poza WD1770/WD1772, ale później i jego dokupiłem), ponadto miałem również stację dyskó do Amigi 500, w 100% identyczną jak ta stosowana w modelu Commodore 1581. W stacji brakowało też kilku śrub i urwany był jeden ze słupków na śrubę, ale wszystko na szczęście udało się posklejać i skrócić.

Stacja dyskó od razu po włączeniu zgłosiła się prawidłowo po próbie dostępu przez C64, co mnie ucieszyło niezmiernie. Jednak po podłączeniu amigowej stacji nie mogłem sformatować żadnej dyskietki DD. Do tej pory nie jestem pewien, czy to lekki defekt stacji, czy też może tak ma być. Podczas dyskusji na kanale #c64 Groepaz twierdził, że jego egzemplarz formatuje jak leci... Rozpocząłem więc poszukiwania oprogramowania, które mogłoby pomóc formatować oraz przenosić dane między PC a stacją dyskó. I tu zaczyna się robić naprawdę ciekawie.

Wśród programów znalazłem:

- Big Blue Reader
- Wolfgang Meser D81Copy - program ten pracuje tylko pod DOS, miewa ponoć problemy, ostatecznie nie testowałem go zbyt długo

oraz pakiet, który używam najczęściej, czyli:

Omniflop / Omnidisk Bingo

Umożliwia on bezproblemowe formatowanie, a potem przenoszenie obrazów różnej maści dyskó na dyskietki 3.5" w stacji PC. Wśród funkcji jest zarówno formatowanie dyskietek dla 1581 jak i zapisywanie, a także odczytywanie obrazów typu D81.

Program ten wymaga instalacji specjalnego sterownika do kontrolera stacji dyskó, który daje bezpośrednią kontrolę nad pracą mechanizmu. Transfer danych z PC na dyskietkę ogranicza się do:

- utworzenia własnego obrazu D81
- sformatowania dyskietki 3.5 cala
- zapisania obrazu D81.

Dla testu można pobrać np. produkcję 69 Positions wydaną przez Vulture Design, która działa ze stacjami 1541/71/81 oraz SD2IEC. Tę kolekcję muzyczną można znaleźć w archiwach na stronie:

csdb.dk

Warto dodać, że w przypadku 1581 trzeba korzystać z dyskietek DD.

Operacja formatowania jak i zapisu trwa po ok. 60 sekund każda. Obraz D81 można przygotować pod emulatorem VICE przy użyciu programu Dirmaster/style (niestety potrafi on robić obrazy błędnie, przez co jakiś czas myślałem że stacja szwankuje) oraz 64Copy, który jest najbardziej godny polecenia, tyle że jest to program dla DOS, więc nie uruchomimy go w Windows x64, aczkolwiek ja sobie radziłem DosBoxem z bardzo dobrym skutkiem.

W zasadzie mogłbym na tym poprzestać, bo cel został osiągnięty, ale jestem człowiekiem dociekliwym, dlatego pozwoliłem sobie jeszcze na zmodyfi-

kowanie mechanizmu FDD-HD (czyli standardowej pecetowej "gęstej" stacji) tak, aby zadziałała w 1581. Nie będę tego opisywał szczegółowo, wystarczy poszukać w Internecie frazy "stacja od peceta do Amigi" - sam korzystałem wiele razy z takich poradników. Niestety front stacji nie pasuje i musiałem plastik odłożyć na bok, ale nie miałem i nie mam zamiaru ciąć plastików tak unikatowej stacji. Stacja HD umożliwia używanie dyskietek HD zamiast DD (należy tylko zakleić jedną dziurkę w dyskietce), oczywiście sformatowanych jako DD - stacja ta po prostu "nie zna" większych dyskietek). Zdobycie nowych dyskietek DD jest prawie niemożliwe, natomiast HD wciąż są (jeszcze) dostępne w handlu.

Po lewej stronie mechanizmu zostało trochę wolnego miejsca, w które trafił przyklejony klejem polimerowym przełącznik do... Jiffy DOS. Używając flasha z uszkodzonej płyty głównej do PC zrobiłem sobie 2x ROM standardowy oraz Jiffy. Oryginalną kość zachowałem, co Wam również polecam. Ze względu na to, że jest to wczesny model 1581 z 1987 roku, był tam stary ROM w pierwszej, zabugowanej wersji. We flashu umieściłem oczywiście wersję nowszą, co jest o tyle istotne, że oryginalna wersja prowadziła do sporadycznej utraty danych.

Rafał Szyja



Stacja dyskietek Commodore 1581 w całej okazałości.



Amiga Hardware Programming

wprowadzenie i odcinki 1-3

„Hello Amiga Coders!” – słowa, którymi Photon/Scoopex zazwyczaj rozpoczyna kolejne części swojego kursu video, opisującego podstawy kodowania intr czy dem na Amigę (w wersji OCS/PAL). Materiał wykładu ukierunkowany jest na wykorzystywanie języka assemblera procesora Motorola MC68000, do bezpośredniego programowania układów specjalizowanych naszego sprzętu, czyli z pominięciem funkcji oferowanych przez system operacyjny. Istnieją na Amidze dema napisane w całości w assemblerze i „100% System Friendly”, my jednak skupimy się na kursie Photona, który obrał takie, niesystemowe podejście. To, że kurs jest naprawdę świetny udowodnił Carrion, znany z polskiej sceny C64 grafik, który na jego podstawie stworzył od zera bardzo przyzwoitą produkcję na A500, a następnie wystawił ją w Amiga EXE Compo podczas RetroKomp / LOAD ERROR 2015, zajmując 5te miejsce.

Jeśli ktoś wyczekiwał jasnego i zrozumiałego „przepisu” jak zacząć kodować na Amigę, od tej chwili może uznać swoje poszukiwania za zakończone i przy odrobinie wysiłku cieszyć się swoją pierwszą scenową produkcją. Artykuł ma na celu ułatwienie zrozumienia kursu Photona, poprzez pełniejsze wyjaśnienie zagadnień poruszanych w poszczególnych odcinkach. Zalecane jest korzystanie z obu materiałów równolegle. Inne aspekty przygody z kodowaniem dem zostały opisane w



numerze zerowym naszego magazynu, przypomnę, więc tylko, że kurs kierowany jest do osób dopiero rozpoczynających naukę i stąd konieczne jest uproszczenie czy bardzo opisowe wyjaśnienia niektórych zagadnień technicznych.

Kurs Photona znajduje się na kanale YouTube:

<https://www.youtube.com/user/ScoopexUs>

Aktualnie 38 odcinków, które omówimy po kolei. Wraz z nabywanym doświadczeniem i wiedzą – kolejne poznawane tajemnice Amigi stają się prostsze i jaśniejsze. Nauka kodowania wymaga czasami przyjmowania pewnych zagadnień bez drobiazgowych wyjaśnień (poniekąd „na wiarę”), – bo szczegóły mocno skomplikowałyby klarowne przedstawienie najważniejszej (w danej chwili) informacji. Czasami lepiej po

prostu przeskoczyć aktualnie mało zrozumiałe fragmenty kursu i iść do przodu, by jednak od czasu do czasu powracać do części poprzednich, i z innego już poziomu przyswoić pominięty materiał.

Dla początkujących - wprowadzenie

W artykule zerowym zapowiadałem, że ominę tłumaczenie rzeczy bardzo podstawowych, bo od tego jest spora ilość kursów internetowych czy książek i każdy zainteresowany tym tematem powinien, samodzielnie się z nim zapoznać (i tak niezbędne będzie częste zaglądanie do opisów technicznych np. rejestrów czy procesora). W rozmowie z kilkoma osobami dowiedziałem się, że warto by zrobić pewne wprowadzenie i to do rzeczy nieraz bardzo podstawowych, które jednak przedstawione bardzo przystępnie i zebrane na początku pomogą w szybkim starcie osobom, które nigdy nie miały styczności z programowaniem niskopoziomowym,



czyli w asemblerze. Zorientowanych w temacie (np. koderów przechodzących z innych platform), proszę o wybaczenie kolosalnego uproszczenia poniższego wprowadzenia i przeskoczenie odpowiednio dalej, a zainteresowanych zupełnymi podstawami proszę o potraktowanie tego wywodu jako obrazowy opis środowiska programowo-sprzętowego, którym będziemy się zajmować.

Ogólnie rzecz ujmując, z punktu widzenia programisty niskopoziomowego na Amidze klasycznej interesują nas: procesor (głównie jego rejestry, dostępne instrukcje oraz tryby adresowania, czyli sposoby posługiwania się instrukcjami), dostępna pamięć (rodzaje RAM: Fast, Chip i FakeFast zwany też Slow), możliwości układów specjalizowanych (Copper, Blitter - które będziemy wykorzystywać do stworzenia efektów wizualnych) oraz narzędzia wspomagające programowanie (assembler AsmOne) i przygotowanie danych graficzno-muzycznych (Protracker, Deluxe Paint, packery, konwertery danych). Do tego dochodzą sprawy związane z Kickstart/AmigaOS, z którymi, mimo, że nie jest nam przy pisaniu dem „po drodze” – to jednak z różnych powodów chcemy się sensownie obejść (na razie tematy dem typu trackmo czy wieloplikowych – pomijamy). Spróbuję opisowo przybliżyć wymienione tematy.

Procesor główny (CPU, ang. Central Processing Unit), to wół roboczy do wykonania programów na naszym komputerze, a demo, to nic innego, jak kolejny program napisany dla Amigi. To, co jest dla nas ważne to, w jaki sposób możemy korzystać z CPU – jakie ma rejestry, jakie instrukcje może wykonywać, w jaki sposób może działać na zawartości pamięci, itp. Motorola 68000 uznawana jest za jeden z najłatwiej „używalnych” procesorów, ze względu na dużą elastyczność i użyteczność swoich instrukcji (czyli dowolność w wykorzystaniu sporej ilości rejestrów - co przyspiesza zarówno pisanie, jak i wykonanie programów) oraz na wygodną pracę na komórkach pamięci (pomocne tryby adresowania). Dodatkowo, procesor ma szeroki wachlarz „rozkazów”, w tym nawet mnożenia i dzielenia (nie było ich w procesorach 8-bit), które choć dość wolne – ułatwiają napisanie np. podstawowej wersji typowej „wektorówki” (grafiki wektorowej, popularne „sześciany”, itp.).

Nasz procesor posiada dwie główne grupy rejestrów, z których będzie korzystał kod, umieszczając w nich dane (np. liczby, znaki ASCII) lub adresy i wykonując na nich obliczenia. Pierwsza grupa to rejestry danych (ang. Data Re-

gisters) ponumerowane od D0 do D7 a druga, to rejestry adresowe (ang. Address Registers) ponumerowane od A0 do A7 (z tym, że A7 zazwyczaj zarezerwowany jest do obsługi tzw. Stosu). Jak widać, rejestry biorą nazwy od swojego głównego przeznaczenia (obróbka danych lub adresów) i jest ich po 8 z każdej grupy. Jak wszystko w asemblerze, numerowanie zaczynamy od zera i ta zasada działa dla numerów rejestrów, bitów, adresów, itp. Wszystkie wymienione rejestry mają wielkość 32ch bitów, czyli mogą przechowywać 4 bajty. Rejestry te można traktować, jako „szczególne komórki pamięci”, do których dostęp jest szybszy i wygodniejszy niż do tradycyjnej pamięci komputera. Dobrym pomysłem jest wykonywanie jak największej ilości pracy na rejestrach, traktując te aktualnie nieużywane, jako podręczne komórki pamięci na tymczasowe wyniki częściowe obliczeń i tak pisać swój kod, by konieczność sięgnięcia do prawdziwego RAMu ograniczać do minimum (kolejna korzyść to ogólnie krótszy kod wynikowy naszego programu).

Bity w rejestrach procesora poukładane są od prawej (bit #0, najmniej znaczący, ang. LSB: Least Significant bit) do lewej (bit #31, najbardziej znaczący, ang. MSb: Most Significant bit). O co chodzi z tym „znaczeniem” danego bitu? Podobnie jak w dacie - „najmniej znacząca” jest ostatnia cyfra w zapisie roku (np. 2016), bo odpowiada za oznaczenie upływu tylko 12 miesięcy, tak druga od końca odpowiada za dziesięciolecie, kolejna za setki lat a „najbardziej znacząca” – za tysiące. Zmiana cyfry na pozycji jednostkowej to tylko rok, zmiana na każdej pozycji coraz bardziej w lewo – powoduje coraz bardziej znaczące zmiany upływu czasu. Podobnie jest z bitami – zmiana jednego bitu na niższych pozycjach powoduje najmniejszą, a więc „najmniej znaczącą” zmianę ogólnej wartości danej liczby, za to zmiana również wyłącznie jednego bitu „po lewej stronie” zapisu – powoduje dużą zmianę wartości liczby.

Rozumiejąc już rozkład MSb-LSb możemy podobnie patrzeć na rejestr i 32 bity w nim zawarte, dzieląc je na 4 bajty, po 8 bitów w każdym. Analogicznie, więc w rejestrze danych możemy wyróżnić bajt najmniej (LSB) i najbardziej znaczący (MSB). LSB ułożony jest tak jak cyfra roku w dacie (po prawej, bity 7-0), MSB – analogicznie, jak cyfra oznaczająca tysiąclecie (bity 31-24). Bity i bajty leżące pomiędzy wymienionymi nie mają specjalnych nazw i ogólnie pojęcia te są raczej opisowe, mają na celu jedynie ułatwić omawianie, na którym bajcie lub od której strony działamy

– wspomagają rozumienie opisu działania algorytmu czy programu.

Teraz czas na sprawę niezwykle istotną - mimo, że nasze rejestry są wszystkie 32-bitowe, wcale nie musimy za każdym razem używać ich w całości, działając na całych 32 bitach, czasem lepiej (szybciej lub wygodniej) jest użyć tylko wybranej części z nich. W zależności od potrzeb, możemy wykorzystać 16 lub tylko 8 mniej znaczących bitów. Jeżeli program potrzebuje wykonywać pracę tylko na 8 bitach – szybciej to zrobi na bajcie, niż na pełnym, 32-bitowym rejestrze. Czasem też łatwiej nam napisać kod myśląc w kategoriach wykorzystania bajtów, niż wartości 4-bajtowych, choć w przyszłości zapewne przyspieszać będziemy działanie naszego kodu poprzez taką zmianę programu, by ładować od razu i obrabiać 4 kolejne bajty – wykonując to jednorazowo i „oszczędzając” na czasie ładowania czy zapisywania poszczególnych bajtów do pamięci, bo każda taka czynność „kosztuje” procesor trochę wysiłku i zabiera czas. Optymalizacja programów (czyli poprawianie, by działał np. szybciej), to temat osobny artykuł.

Wracając do rejestrów, można działać na nich, wykorzystując 1, 2 lub 4 bajty, czyli odpowiednio 8, 16 i 32 bitów. Te wielkości mają swoje tradycyjne nazwy i są to odpowiednio bajt (8 bitów), słowo (16 bitów) i długie słowo (32 bity). Kiedy piszemy kod i chcemy poinformować program asemblerujący (np. AsmOne), że chodzi nam o działanie na danych o rozmiarze bajtu – zapisujemy to pisząc nazwę instrukcji, np. dodawania (instrukcja ADD), a potem, po kropce, nasz pożądaný rozmiar: „b” dla bajtu, „w” dla słowa (ang. Word) i „l” dla długiego słowa (ang. LongWord). W wyniku tego instrukcja „add.b #20,d0” doda 20 tylko do dolnego bajtu rejestru D0, instrukcja „ADD.W #777,D1” doda 16-bitowe słowo do dolnych 16 bitów (2 bajtów) w rejestrze D1, a „ADD.L #\$1F0015ab,d2” doda zapisaną heksadecymalnie (czyli w systemie szesnastkowym) dużą liczbę \$1F0015ab (dziesiętnie jest to 520099243) do rejestru D2.

Tak samo możemy z rejestru odczytać bajt, słowo lub długie słowo – w zależności od tego, co akurat potrzebujemy i nie ma żadnego znaczenia, czy poprzednia instrukcja działała tam w takim samym rozmiarze, czy na innym – to my i nasz program decyduje, jaką instrukcję potrzebujemy wykonać i na jakim rozmiarze danych. Co więcej, jeśli potrzebujemy wykonać coś na rejestrze danych – nie ma znaczenia czy wykorzystamy do tego D4 czy D1 – to wła-



śnie, wraz z możliwością traktowania rejestrów, jako bajt/słowo/długie słowo, powoduje, że możemy pisać swoje programy „jak nam się podoba” i mamy dużą dowolność, jak to sami sobie to zorganizujemy. Jest to cecha charakterystyczna MC680x0, bo np. wiele instrukcji procesorów 8-bitowych czy rodziny Intel 80x86 wymagają nieraz umieszczania wartości w konkretnych rejestrach, by instrukcja zadziałała poprawnie i na właściwych danych. Nasz CPU daje nam pełną swobodę i tylko od nas zależy jak to wykorzystamy.

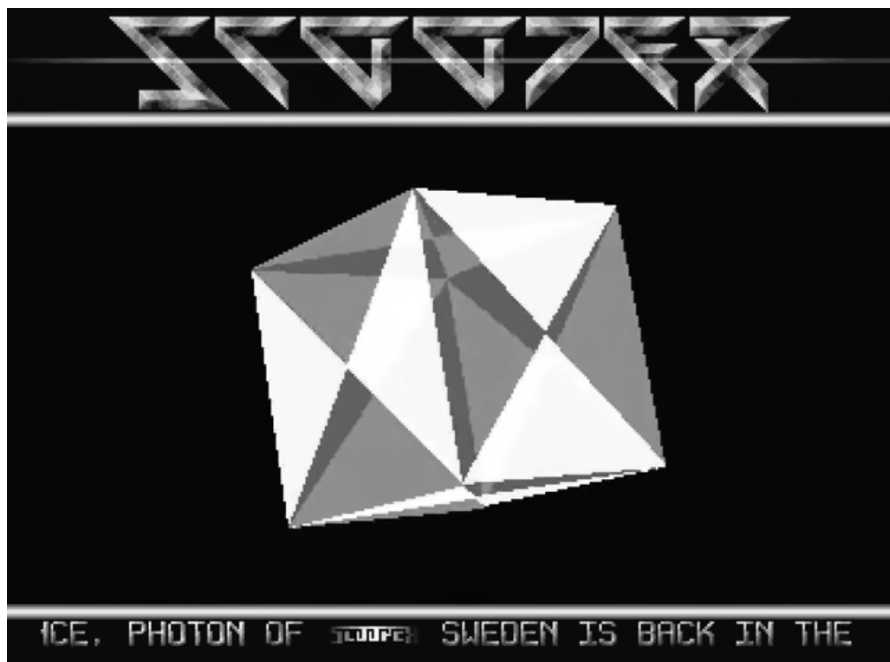
Działając w rejestrze danych na bajcie czy słowie – zmieniamy tylko bity należące do tego rozmiaru. Jeśli przekroczymy jego zakres, np. w bajcie, dodając jeden do będącej już w rejestrze wpisanej liczby 255 (= \$ff = %1111 1111) będącej największą, jaką możemy zapisać w tym rozmiarze danych – bity od 7 (MSb w tym bajcie) do 0 (LSb) nam się wyzerują, bo „przekręci się licznik”. Gdybyśmy używali większego zakresu, a więc co najmniej 16-bitowego słowa – został by użyty kolejny bit (czyli bit #8, numerując od zera) i otrzymali byśmy liczbę 256 (= \$100 = %0000 0001 0000 0000). Tutaj jednak zakres nam się skończy i poniekąd stracimy część wyliczonego wyniku.

Na szczęście nie jest to do końca prawdą i takie działanie możemy rozpoznać, dzięki wyznaczonym i osobnym bitom w procesorze, które służą właśnie do oznaczania takich „specjalnych sytuacji” i są odpowiednio ustawiane bądź kasowane – po każdej wykonanej przez procesor instrukcji. Bity (zwane też flagami, znacznikami) są umieszczone w wydzielonym i dodatkowym rejestrze procesora, zwanym Rejestrem Statusowym i mogą wspomagać podejmowanie decyzji w programie lub kierować jego przebiegiem. Kończąc opis głównych rejestrów trzeba jeszcze wspomnieć, że rejestry adresowe nie mają możliwości przesłania do nich pojedynczego bajtu czy wykonania na nich jednobajtowych operacji – tu używamy rozmiaru słowa lub długiego słowa.

Podając powyżej trzy przykładowe instrukcje dodawania (ADD) zapisałem je stosując małe i wielkie litery. Piszac tak, chciałem pokazać, że można zapisywać je dowolnie, nawet mieszając sposób zapisu – większość assemblerów bez problemu sobie z tym poradzi. W trzecim przykładzie („ADD.L #\$1F0015ab,d2”) użyłem dwóch ciekawych znaków, na początku podanej liczby 32-bitowej: „#\$xxxxxxx”. Znak '#' informuje assembler, że chodzi nam o liczbę (a nie o adres) a znak '\$' ozna-

cza, że liczbę zapisałem w systemie szesnastkowym, najbardziej użytecznym i najłatwiejszym w użyciu, jeśli programujemy w assemblerze. Oczywiście komputer pracuje wewnątrz na systemie binarnym i taki zapis możemy zastosować w instrukcji, jednak zazwyczaj

asemblujący (np. AsmOne w dowolnej wersji, polecenie „? wartość/działanie”). Assembler, program asemblujący, AsmOne... Ogólnie wiadomo, że: „amigowi koderzy kodują dema w assemblerze i piszą w AsmOne, asemblując kod programem asemblującym”... Masa-



jest on długi i nieczytelny. Warto go jednak używać, jeśli chcemy pokazać dokładnie, które bity planujemy daną instrukcją ustawić (zapalić), a które wyzerować (zgaścić) – tego typu zapis przyda nam się, gdy zaczniemy wpisywać wartości do układów specjalizowanych, takich jak Copper, przy przerwaniach czy obsłudze DMA (o wszystkim będzie później), gdzie czasem trzeba zmienić tylko wybrane bity. W zapisie zero-jedynkowym, znakiem '%' poprzedzamy liczbę binarną (np. %10110101 to dziesiętnie liczba 181, a szesnastkowo \$B5).

O systemach bin/dec/hex można przeczytać w każdym wstępie do programowania w assemblerze, my musimy tylko wiedzieć, że niezależnie od systemu zapisu chodzi zawsze o tą samą liczbę i to od nas zależy, która forma w danym momencie jest dla nas wygodniejsza. Zapis dziesiętny nie wymaga oznaczania jego użycia żadnym specjalnym znakiem i zazwyczaj stosuje się go dla małych liczb, powyżej wartości bajtu zdecydowanie lepiej sprawdza się system heksadecymalny. Wymienione systemy liczbowe warto znać, by płynnie i zamiennie je stosować, jednak nie musimy przeliczać ich ręcznie – zrobi to za nas każdy trochę bardziej zaawansowany kalkulator lub dobry program

kra? :) Cóż, to, co na pierwszy rzut oka (i ucha) brzmi zupełnie niezrozumiale – ma jednak sens. Assemblerem zwyczajowo nazywamy zarówno kod programu, napisany w języku niskopoziomowym, czyli bliskim sprzętowi i łatwo przetłumaczalnym na instrukcje wykonywane przez procesor, jak i sam program, który dokonuje takiego tłumaczenia - w naszym przypadku będzie to najbardziej popularny na scenie (choć nie jedyny), legendarny AsmOne. Tak naprawdę, CPU nie wykonuje programu zapisanego w assemblerze, tylko odpowiadający mu binarny „kod maszynowy”, czyli rozpoznaje pewne ciągi bitowe, rozkodowuje sobie w środku zawartość pobranych z pamięci rozkazów i na ich podstawie – działa.

Gdyby człowiek chciał pisać bezpośrednio w kodzie maszynowym – powinien składać w pamięci RAM wartości liczbowe odpowiadające instrukcjom procesora – to dopiero byłoby uciążliwe zadanie (choć czasem tak się robi i nie jest to aż takie super trudne). Żeby jednak zaoszczędzić tak żmudnego zajęcia – powstały assembly, czyli programy „asemblujące” (ang. assemble – montować, składać), które tłumaczą program napisany przez człowieka i składają z niego liczbowe odpowiedniki



rozkazów, zrozumiałe dla CPU. Narzędzia typu assembler potrafią o wiele więcej (disasemblować pamięć, debugować kod, generować tablice sinusów, odczytywać/zapisywać sektory na dyskietce), ale głównym ich zadaniem jest właśnie przetworzenie zapisu programu zrozumiałego przez nas - na język maszyny. To, że nazwy assembler i język maszynowy bywają używane zamiennie (choć nie do końca prawidłowo), jest wynikiem tego, że każda instrukcja zapisana przez programistę odpowiada dokładnie jednej instrukcji rozumianej przez procesor, czyli człowiek musi myśleć i zapisywać program „w języku maszyny”, a program assemblera już tylko zmienia formę zapisu na zrozumiałą i gotową do wykonania przez CPU. Człowiek zapisujący kod dla maszyny i w jej języku nazywany jest koderem, a programowanie w tym języku – nazywamy kodowaniem.

Co do samego AsmOne – został on napisany i wydany komercyjnie przez Promaxa z grupy Kefrens, jednak sprzedawany był tylko wersjach 1.0, 1.01 i 1.02. Od 1992 roku dalszym jego rozwojem zajmuje się grupa The Flaming Arrows (T.F.A.) i jest to produkt uwolniony, obsługujący całą rodzinę MC680x0, koprocesory matematyczne a nawet PowerPC. Aktualna stabilna wersja to 1.48 (wydana w 2002r.), choć miewa problemy z debuggerem („zwiechy” pod UAE) i dlatego polecam każdą inną, starszą wersję – np. 1.20 (wersje wyższe wymagają ks2.0+) lub ostatnią wersję testową 1.49-RC2 (wydana w 2008r.). Tak naprawdę konkretna wersja nie ma znaczenia, bo używane przez nas (do potrzeb kursu Photona) funkcje programu dostępne są w każdej z nich, więc niech każdy dobierze sobie taką, jaka mu się podoba.

Instalacja programu nie sprawia problemów, praktycznie wystarczy rozpakować archiwum i od razu korzystać ze standardowych ustawień – czasem tylko może być wymagane skopiowanie fontów czy req/reqtools/mathtrans.library do odpowiedniego folderu na naszej partycji systemowej oraz reboot, po czym wszystko powinno być gotowe. Polecam świetny opis AsmOne w postaci książki Adama Doligalskiego, który przedstawił nie tylko sam program, ale i wiele praktycznych rad czy przykładów oraz nowy kurs autorstwa Asmana/WT, niedawno rozpoczęty na portalu PPA.

Ok, rozumiemy już trochę jak działa CPU, co to jest rejestr, jak ułożone w nim są dane. Wyjaśniamy sobie zamienne systemy liczbowe, układy bitów, grupy rejestrów, odpalamy assembler,

zapalamy i gasimy flagi w CPU. Wiemy, że do pracy będzie potrzebny nam assembler, że w nim będziemy pisać kod, który trafi do procesora – tylko po co? Co to ma wspólnego z kolorowymi efektami na ekranie? Plazmami, wektorami, twisterami? Jak te nudne liczby czy pojedyncze bity wpływają na to, co widzimy w demach na ekranie, – po co nam to wszystko, gdzie w tych suchych faktach „gotuje się magia”? Cóż, czas na przyjrzenie się nieco amigowemu tzw. układowi specjalizowanemu czyli słynnym Custom Chips.

Wspomniane wyżej części składowe, tak charakterystyczne dla naszej Przyjaciółki, to „układy przygotowane na zamówienie”, specjalnie dla Amigi, by wspierać procesor (a nawet go wyręczać) w wykonywaniu czasochłonnych zadań. Te zadania to np. wyświetlanie wielokolorowej plazmy (Copper), kopiowanie dużych obszarów ekranu czy rysowanie linii (Blitter). Kiedy w demach OCS te rzeczy widać na ekranie – zazwyczaj zamiast CPU pracują właśnie specjalne układy, które zostały zaprojektowane do tego, by procesor przekazał im tylko, co mają zrobić (wysterować je, zaprogramować), a one same wykonały resztę. Teoretycznie, w trakcie ich działania – procesor może robić „coś innego” - czy to w 100% jest prawda - to temat na dość skomplikowany artykuł. Na razie możemy wyobrazić sobie, że te układy robią to szybciej niż procesor, więc niech CPU nimi zarządza i dopóki działają, niech się o ich zadania nie martwi lub przygotowuje kolejne do wykonania.

Tak, więc magia OCS-owej Amigi dzieje się głównie w układach specjalizowanych i to dzięki nim oglądamy efekty w demach, a MC68000 jest potrzebny po to by kierować tym, co ma dla nas wykonać reszta hardware Amigi. Pisząc kod dla procesora (w assemblerze) – podajemy mu sposób, w jaki ma sterować/zarządzać resztą układów, by osiągnąć to, co zamierzaliśmy – błysk na ekranie, scroll, wektorową bryłę, itp. Do tego trzeba też wiedzieć, że układy specjalizowane Amigi, choć są dość wydajne i potężne jak na sprzęt zaprojektowany w połowie lat 80ch poprzedniego wieku, to same z siebie są dość mało „inteligentne” i wymagają ciągłego „popychania do roboty” - właśnie przez procesor. Dopiero odpowiednio zarządzane przez CPU, mogą pokazać swoje duże możliwości, a że zarządzać podwładnymi można zarówno źle, jak i dobrze (i takie efekty osiągać) – to od kodera zależy, czy pisząc kod dla procesora wykorzysta to co jest do jego dyspozycji w sposób lepszy, czy gorzy.

Jak programować układy specjalizowane? Z ogólnikowych opisów OCS wiemy, że układy te to Agnus, Paula i Denise, możemy rysować ich schematy blokowe, itd. Od strony programowej, wcale nie musimy dokładnie wiedzieć (choć dobrze by było), gdzie mieści się np. Blitter, bo układy te „ukazują się” jako specjalne komórki pamięci (tzw. rejestry sprzętowe, dające dostęp do hardware Amigi), tyle że pod precyzyjnie wyznaczonymi adresami oraz z przepisem, jak właściwie z nich korzystać, bo nie zawsze możemy do nich dowolnie coś zapisać lub odczytać. Niektóre rejestry są tylko do odczytu, byśmy mogli sprawdzić aktualny stan wykonania naszych zleconych zadań, np. czy są już zakończone poprzednie działania (wyczyszczenie kawałka ekranu) i czy możemy „zatrudnić” ten sam układ do kolejnej pracy (skopiowania w to miejsce kawałka obrazka). Inne rejestry są tylko do zapisu i w nich składowy odpowiednio przygotowane „zlecenia”, a odczyt może spowodować zwrócenie bezsensownych danych lub nawet zawieszenie się komputera.

Są też i rejestry zarówno do odczytu jak i zapisu – by wiedzieć, który jak pracuje, należy sprawdzić to w dokumentacji Amigi i tutaj niezbędne stanie się przeglądanie, np. „amigowej Biblii”, czyli AHRM: Amiga Hardware Reference Manual, która podaje opis działania i sposób użycia każdego rejestru. Powstanie kolejne pytanie, – jak więc sterować tymi układami specjalizowanymi? Otóż właśnie umieszczając w rejestrach zawarte w naszym programie lub przygotowane przez obliczenia wartości (liczby, czasem pojedyncze bity) – przy pomocy procesora. Nasz kod, napisany dla CPU, będzie wpisywał do rejestrów hardware Amigi zaprojektowane wartości, a układy te, na podstawie wpisanych do nich liczb i ustawień – wykonają wymyślone przez nas zadania.

Tak właśnie odbywa się cała magia – sprzęt czeka w gotowości, by kod napisany przez kodera powiedział mu, co i w jakim momencie ma się wykonać, a efekty tego działania zobaczymy na ekranie. Warto przypomnieć też, że kodując z pominięciem systemu, oprócz wyłączenia wielozadaniowości omijamy również i biblioteki - musimy, więc ze wszystkim radzić sobie sami. Dlatego też to my powinniśmy zadbać o komplet wartości we wszystkich wymagających tego rejestrach – inaczej wykonanie naszego zadania pójdzie zapewne „w las”, bo istniejące w rejestrach przypadkowe wartości pokierują sprzętem w nieprzewidywalną stronę. W samych bibliotekach nie ma nic złego, stworzone były



one by spełniać możliwie dowolne oczekiwania programisty i zostały napisane bardzo ogólnie oraz nadmiarowo. Ten niepotrzebny nam kod powoduje, że maszyna pracuje po prostu zbyt wolno, jak na potrzeby zrealizowania typowego demoscenowego efektu. My, zarządzając dostępnym hardware, wykonujemy tylko to, co jest konieczne i dlatego jesteśmy szybsi. Pomijane przez nas funkcje biblioteczne i tak w końcu do wykonania swoich zadań wykorzystają te same rejestry sprzętowe, dlatego nasza przewaga jest w pominięciu wykonania czynności nam niepotrzebnych i jeśli wszystko zrobimy jak należy, ani samej Amidze, ani też systemowi operacyjnemu krzywdy nie zrobimy a nasz wymyślony super efekt wykona się o wiele szybciej niż analogiczny, wykonany przy pomocy wolniejszych (i ograniczonych) funkcji bibliotecznych.

Procesor, układy specjalizowane, funkcje biblioteczne – o czym zapomnieliśmy? O pamięci! :) Na szczęście to temat dość prosty, więc szybko nadrobiamy i przechodzimy w końcu do działania. Pamięć komputera to układ elektroniczny, który służy do przechowywania w nim różnych danych: liczb, wartości, bitów, bajtów, słów czy długich słów, czyli kodu programu, grafiki, sampli muzycznych – słowem wszystkich danych, z których korzysta komputer. Wszystkich, czyli jakich? No właśnie wszystkich, bo w zależności, co (CPU, układ specjalizowany) pobiera tę samą wartość i z tego samego miejsca – może być, czym innym, bo to „układ używający” daną liczbę – interpretuje, czym ma ona być i co z nią zrobić. Jeśli pobierze to procesor i wykorzysta (wykonując swój program) daną wartość, jako element swoich obliczeń, to liczba ta będzie „daną matematyczną”. Jeżeli CPU wykorzysta tą samą liczbę, jako „wartość konfiguracyjną” czynność do wykonania przez układ specjalizowany – może Blitter narysuje nam coś, albo Copper wyświetli kolor. Jeśli pobierze to Blitter, to może wektorowa ściana wypełni się w jakiś sposób (np. wzorem) lub też może Copper wyświetli ciekawą plazmę.

A może ta wartość to litera tekstu? Być może – to zależy wyłącznie od kodera, jak dane przygotowuje i porozmieszcza w pamięci oraz jego pomysłu, czym w danym momencie konkretna wartość ma być. Innymi słowami – to koder programuje komputer, by daną wartość, umieszczoną w pamięci, wykorzystał w konkretny sposób. Pamięć to tylko miejsce składowania liczb, a to CPU i układy specjalizowane zdecydują (na

podstawie przygotowanego kodu), czym te wartości się staną. Obrazek czy muzyka – to też tylko zbiór odpowiednio poukładanych liczb, które naberają sensu tylko w momencie ich właściwego wykorzystania. By swobodnie używać pamięć w programie trzeba wiedzieć, że można odwołać się do każdej jej komórki (bajtu) opisanej przez adres, czyli jej numer w kolejności „ułożenia” w układzie elektronicznym.

Słowo „adres” jest często używane by opisać np. początek obrazka, albo tablicy kolorów w jego formacie danych. Adres może być początkiem miejsca w pamięci, gdzie wczytane zostały sample modułu lub miejscem, gdzie w CopperList (programie dla Coppera) trzeba wpisać inny adres, by móc wyświetlić pikselowany rysunek lub kolorową linię. Adres ma ogromną ilość zastosowań, w zależności, co chcemy osiągnąć, jednak zawsze koniec końców jest to numer konkretnej komórki pamięci i to od nas będzie zależało, jak sprawnie (czyli z jaką wiedzą) będziemy się nim posługiwać. Adresy można dodawać do siebie, zwiększać czy zmniejszać automatycznie lub na żądanie, porównywać, obliczać odległość jednego adresu od drugiego, itp.

Adresy mogą być stałe lub tymczasowe, w obszarach ROM, ChipRAM, FastRAM, w obszarze rejestrów sprzętowych. Adresy nieraz należy wpisać do Coppera lub Blittera, żeby układy specjalizowane wiedziały, na jakich danych i w której części pamięci mają pracować. Adres może być bezpośredni lub pośredni, czyli przygotowany poprzez pewne wyliczenia. O tym, co możemy robić z adresami decydują dostępne tryby adresowania, czyli sposoby, jakimi procesor może coś zadziałać na danych. Tryby adresowania będą powoli wprowadzane przez Photona w trakcie ich używania, ale zdecydowanie polecam, choć wstępne doedukowanie się czytając jakiś opis MC68000, naprawdę ułatwi to zrozumienie przedstawianych przykładów. Nie chodzi wcale o wrycie ich na pamięć, a tylko wstępne zarysowanie sobie ich potencjalnych możliwości użytkowych.

Ostatnią informacją, którą przed rozpoczęciem kursu należałoby poznać, jest różnica między ChipRAM, a FastRAM. Ogólnie wiemy, że nazwy same już trochę opisują ich przeznaczenie. Pamięć typu Chip to główna pamięć, bo występująca w każdym modelu Amigi i to od adresu zerowego, czyli od samego początku „zamontowania” pamięci. Ta pamięć wystarczy do wszystkich głównych

działań. Nazwa „Chip” pochodzi od tego, że to właśnie z tego typu mogą korzystać nie tylko procesor, ale i układy specjalizowane. Tam właśnie muszą znajdować się dane, na których układy mogą pracować i tylko tam składać wyniki swoich działań. Próba wymuszenia pracy układów na poza pamięcią Chip – skończy się zapewne rebootem.

Problem z pamięcią Chip jest taki, że programy umieszczone w niej wykonują się wolniej, a to, dlatego, że układy specjalizowane mają pierwszeństwo w dostępie do niej i CPU nieraz musi czekać, aż będzie miał możliwość kontynuować swoje zadania. Taka sytuacja nie występuje w przypadku pamięci Fast, bo układy specjalizowane nie mają tam dostępu i CPU może działać ze swoją normalną prędkością. Pamięć leży za Chip, czyli w innej przestrzeni adresowej. Z powodu jej prędkości, kod wykonujący ważne obliczenia i przygotowanie danych najlepiej jest umieszczać właśnie tam, o ile taką pamięć wykryjemy, bo nie możemy być pewni, czy w danym komputerze ta pamięć (opcjonalna) jest zainstalowana.

Jak wiadomo, tego problemu nie ma z pamięcią Chip, która jest zawsze dostępna. Pamięci Fast nie należy jednak mylić z tzw. Slow lub FakeFast, bo, mimo, że układy nie mają tam dostępu – CPU też nie może do końca rozwinąć tam skrzydeł i działa podobnie wolno jak w Chip. Pamięć leży w jeszcze innym obszarze adresowym niż dwie poprzednie, a popularność swoją zawdzięcza głównie niskiej cenie. Rozszerza ona oczywiście możliwości komputera, ale wykorzystywana bywa zazwyczaj, jako rodzaj „ram-dysku”, np. do przechowywania spakowanych kolejnych części dema.

Czas w końcu uruchomić AsmOne i zacząć właściwą zabawę!

Amiga Hardware Programming 1

Use the Assembler
and wait for left click

Praca z AsmOne zaczyna się od wyboru typu i ilości pamięci, jaką chcemy, by assembler zarezerwował na pliki robocze naszego dema, czyli wczytany z dysku kod (który będziemy edytować i rozwijać), dodatkowe pliki z definicjami przydatnych wartości, adresów czy nazw (tzw. include) oraz potrzebne nam pliki binarne. Skoro demo będzie głównie pracować na pamięci typu Chip, to



lepiej te rzeczy trzymać gdzieś poza naszym docelowym obszarem działania. Jeśli dostępna jest większa ilość pamięci Fast, rozsądniej będzie właśnie tam zagospodarować ten obszar. Co do wielkości – należy jakoś ogólnie oszacować sobie ile aktualnie nasz projekt może zajmować i nie przesadzać z wielkością, bo im więcej pamięci przeznaczymy na te cele – tym mniej zostanie systemowi na inne programy, a zazwyczaj obok assemblera mamy w tle uruchomione programy graficzne, konwertery czy packery. Dla małych programów wystarczy nawet 10kB, ja zwykle wpisuje 100-200kB (ze względu np. na player muzyki) i tyle na początek mogę polecić. Następnie warto zaznajomić się z programem, jak coś wczytać lub zapisać, jak uruchamia się monitor pamięci, debugger i edytor. Polecam uruchomić tą samą funkcjonalność raz z menu, a raz z linii poleceń (ta druga wersja bardzo usprawnia i przyspiesza pracę). Wejście do edytora oraz przerywanie działania różnych funkcji programu, powrót z monitora pamięci czy debuggera odbywa się przy pomocy klawisza Esc (Escape), warto, więc sobie to przetrenować.

Celem tego odcinka jest napisanie fragmentu kodu, który występuje praktycznie w każdym demie, a więc pętli oczekiwania na przyciśnięcie przez użytkownika lewego przycisku myszy. Praktycznie każde intro czy demo powinno po naciśnięciu LMB zakończyć swoją pracę i powrócić do miejsca, z którego zostało uruchomione – nieraz organizatorzy demo party wręcz wymagają tego i wpisują taki punkt do regulaminu konkursu. Pętla ta przyda się nam również przy pisaniu i testowaniu kolejnych efektów, bo nieraz będziemy chcieli poczekać odpowiednio długo (aż efekt „rozwinie się” lub odpowiednio zaprezentuje) i przerwać jego pracę w odpowiednim momencie, np. po to, żeby sprawdzić czy efekt działa poprawnie lub zanotować jego najciekawsze wartości, które finalnie zostaną użyte w demie. Naciskając Esc przechodzimy do edytora „tekstu programu” i razem z Photonem zaczynamy pisać kod. Warto zwracać uwagę na wszystkie wprowadzane przez niego ręcznie komendy, bo nie wszystkie są wyjaśniane (wynika to z automatyzmu pracy Autora kursu).

W swoim wprowadzeniu do tego, co zamierzamy napisać, Photon używa określenia „busy-loop/busy-wait-loop” i na tej zasadzie będzie pracował nasz pierwszy mini-program. Tak jak sugeruje opisowa nazwa, czynność sprawdzania czy został naciśnięty LMB będzie wykonywana w najprymitywniejszej pętli typu: „sprawdź czy wciśnięty LMB,

sprawdź czy wciśnięty LMB, sprawdź czy wciśnięty LMB, ...” lub inaczej: „oczekuj na wynik, robiąc tylko to - test za testem, aż do wykrycia wciśniętego LMB”. Przy większych projektach takie podejście jest oczywiście bez sensu, bo procesor jest niepotrzebnie „przetrywany” przez tak proste zadanie, które można by wykonywać np. raz na 1/50 sekundy, czyli raz na tzw. „ramkę” (ang. frame, jedno pełne wyświetlenie całego ekranu, które na Amidze w wersji PAL odbywa się 50 razy na sekundę, zapamiętajmy to). Oczekując na LMB w sposób „cały czas zajęty”, blokujemy sobie możliwość wykonania innych potrzebnych zadań. Tutaj jednak, gdzie celem jest programik ćwiczeniowy – można tak postąpić, bo oczekiwanie na LMB jest jedynym planowanym przez nas zadaniem dla tego programu. Pętla tego typu warto jednak poznać, bo przy krótszych zadaniach (np. oczekiwanie na początek lub koniec pojedynczej linii rysowanej na ekranie) taka forma może mieć już całkiem spory sens.

Nasz kod zaczynamy od tzw. etykiety (czy też „labelki”, od ang. label). Te dwie nazwy są niczym innym, jak zrozumiałym dla człowieka „tytułem” fragmentu kodu. Nadając im sensowne, opisowe nazwy, ułatwiamy sobie sami pracę z programem, bo przy kilku(-nastu) tysiącach linii kodu (a tyle miewają dema) – dobry opis części kodu (wraz z jak najbogatszymi komentarzami) jest koniecznością. Etykieta „waitmouse:” od razu sugeruje, czym dany fragment będzie się zajmował. W nazwach etykiet proponuję używać też wielkich liter czy znaku ‘_’ (bo nie można używać tu spacji), co jeszcze bardziej poprawi czytelność „nazwy” (np. „WaitMouse:” czy „Wait_Mouse:”). Zwyczajowo, etykieta powinna zaczynać się w pierwszej kolumnie ekranu (instrukcję zaś po jednej lub więcej spacji czy tabulatorze) i kończyć się dwukropkiem (‘:’). AsmOne dopuszcza też zrezygnowanie z ‘:’ oraz inne ustawienia (np. rozróżnianie małych i wielkich liter w zapisie labelki) – wszystko to jest do skonfigurowania w zakładce „Preferences” i polecam dopasowanie do siebie, po zapoznaniu się z podstawami programu (np. ja zawsze włączam numerację linii – w celu sprawniejszego poruszania się po własnym kodzie).

Nadanie etykiety ma również jeszcze kolejny pozytywny aspekt – podczas asemblacji każda labelka zostanie zamieniona na adres w pamięci, pod którym przetłumaczony kod wynikowy zostanie „ułożony”. Proponuję tutaj mały eksperyment: kiedy cały program tego odcinka zostanie już wpisany i zapisany na dysku, uruchomcie AsmO-

ne trzy razy ponownie, za każdym razem wybierając inny rodzaj pamięci na starcie (Allocate Fast/Chip/Public). Następnie zasemblujcie wczytany kod komendą ‘a’ (od ang. Assemble) i bez uruchamiania go wpiszcie komendę „d waitmouse”. Polecenie to zdisasemblije (przetworzy liczbowy kod maszynowy na zrozumiałe przez człowieka instrukcje) fragment pamięci rozpoczynający się od etykiety „waitmouse:”. Zwróćcie uwagę na pierwszą od lewej dużą liczbę – są to adresy w pamięci każdej instrukcji, jakie trafiły tam w wyniku asemblacji naszego programu.

Jeśli wybraliście pamięć Fast, adres może wyglądać np. tak: 101B2E30. Jeśli wybierzeć pamięć Chip, będzie to np. 2F540. Dlaczego? Otóż obie liczby to adresy zapisane szesnastkowo (disassembler pomija znaczek ‘\$’ przed liczbą, oszczędzając kolumnę na ekranie), z tym, że ta duża to adres w obszarze pamięci FastRAM, która zaczyna się od obszaru \$200000 i ciągnie przez kolejne kilka lub kilkadziesiąt megabajtów, a ta mniejsza, to adres z obszaru ChipRAM, który zaczyna się od adresu \$000000 i kończy (przy zainstalowanych 2MB) na adresie \$1fffff (czyli tuż przed początkiem FastRAM). Przykład ten pokazuje, że dzięki etykiетom, możemy się dokładnie przyjrzeć się gdzie oraz jak nasz kod został rozłożony i to, w której pamięci (o ile sami nie pokierowaliśmy tym procesem). Sprawdzanie jak wygląda nasz kod, czy wyniki jego działania np. na buforze w pamięci (który też będzie miał swoją własną etykietę i możliwość odszukania w ten sposób) to jedna z najszybszych i najczęstszych form sprawdzania i debugowania naszego kodu.

Nadszedł w końcu czas, żeby napisać kod, który „coś robi”, a mianowicie sprawdza, czy LMB jest wciśnięty, czy puszczony. Okazuje się, że możemy to sprawdzić odczytując jeden bit (numer 6) w rejestrze pod adresem \$bfe001, który to rejestr udostępniany jest nam przez układ 8520 CIA (Complex Interface Adapter). Dlaczego bit #6 i pod tym właśnie adresem, bo tak zaprojektowali to twórcy Amigi. Skąd wiedzieć, jaki bit i skąd odczytać? Trzeba przeglądać rozdziały książek opisujących hardware Amigi. Tam też dowiemy się, że bit odpowiedzialny za stan LMB jest „active low”, czyli jego normalny stan (LMB puszczony) to bit zapalony (=1), a gdy LMB jest naciśnięty – bit zostaje skasowany i poprzez odczytanie bitu #6 w tym rejestrze, jako zero – wykrywamy naciśnięcie klawisza. Dlaczego używać instrukcji BTST (Bit Test)? Trzeba choć trochę zapoznać się z jakimkolwiek opisem MC68000 albo podejrzeć jak to ro-



bią inni, w opublikowanych gdziekolwiek szkoleniowych fragmentach kodu, a na sieci jest tego sporo. Mało jest języków programowania takich, że wystarczy przeczytać 1 książkę czy przerobić 1 kurs, by radzić sobie bez problemu w większości sytuacji. Nauka i późniejsze programowanie to ciągle przeglądanie różnych materiałów i zapamiętywanie lub wynotowywanie przydatnych fragmentów.

Wiedząc już skąd i jak odczytać aktualny stan LMB, możemy skonstruować naszą pętlę oczekującą. Jak pisałem wcześniej, wykonanie każdej instrukcji może mieć wpływ na specjalne bity (flagi, znaczniki) w CPU, w Rejestrze Statusowym. Jeśli jakkolwiek ostatnio wykonana operacja w procesorze dała wynik zero (np. odejmowanie), to w Rejestrze Statusowym będzie ustawiony bit Z czyli flaga Zero (skojarzenie: flaga wciągnięta na masz do samej góry). Odczyt wartości pojedynczego bitu też może dać wynik zerowy i w związku z tym zapali się odpowiedni bit statusu. Dlatego wykonanie instrukcji testującej bit LMB ustawi nam odpowiednio flagę Zero i kolejna instrukcja (w naszym przypadku jest to instrukcja „skoku wa-

czyta niski ('0') stan – rozkaz BNE stwierdzi, że skok jest już niepotrzebny i pozwoli CPU wykonać następną instrukcję (leżącą za BNE w pamięci), czyli w naszym przypadku będzie to RTS (ReTurn from Subroutine – powrót z podprogramu / funkcji / procedury). Taki powrót zazwyczaj występuje, gdy nasz fragment kodu stanowi często powtarzający się i wywoływany podprogram, który zamiast być wpisywany za każdym razem, gdy jest potrzebny, można sobie wyodrębnić i trzymając gdzieś z boku wywołując, kiedy jego praca będzie nam potrzebna. Po takim wywołaniu (rozgałęzieniu) trzeba jakoś wrócić do tej części programu, która dany podprogram „zatrudniła” i do tego właśnie służy instrukcja RTS.

Pisząc i testując nasz program pod kontrolą AsmOne musimy wiedzieć, że assembler uruchamiając nasz kod - traktuje go, jako własny podprogram, więc żeby „nie zawiesić” naszego narzędzia – po wykonaniu naszego zadania trzeba do niego „wrócić” (chyba, że lubimy częsty reboot lub świadomie zaprogramujemy coś innego, np. doczytanie kolejnej części dema z dysku).

programu dla CPU w pamięci (a jest to czynność na tyle skomplikowana, że wymaga zazwyczaj 2ch „przebiegów” assemblera po naszym kodzie). Jeżeli wszystko zrobiliśmy poprawnie (zapis instrukcji, definicje danych, podaliśmy prawidłowe ścieżki doczytywanych plików), po 2ch przebiegach asemblacji (Pass 1.., Pass 2..) powinno ukazać się nam upragnione „No Errors” (jeśli jest inaczej, AsmOne podpowie w której linii i mniej więcej w czym jest problem). Teraz możemy zajrzeć sobie pod etykietę „waitmouse:”, przyjrzeć się jak kod wynikowy (zasemblowany) wygląda, możemy ręcznie wprowadzić poprawki, ustawić testowe wartości w rejestrach czy pamięci, itp. Najczęściej jednak chcemy sprawdzić czy to co napisaliśmy zadziała tak jak chcieliśmy – wykonujemy więc komendę 'j' (od „jump”) i AsmOne „skoczy wykonać” nasz program (skąd wie, pod jaki adres i gdzie zacząć – to już temat na później, na razie przyjmijmy, że do pierwszej linii naszego kodu).

Co widać na ekranie, a co naprawdę się wykonało? Widać może niewiele, bo nasz program miał zaledwie 3 instrukcje. Skoro napisaliśmy pętlę oczekującą na LMB i uruchomiliśmy ją – to aktualnie wykonywane jest „oczekiwanie na LMB” (zauważmy też, że AsmOne wydaje się jakiś zablokowany). Klikamy przycisk, zeruje się bit odpowiedzialny za stan LMB, instrukcja BNE przestaje zapętlać jego testowanie, CPU wykonuje kolejny rozkaz (RTS) i wracamy z powrotem do odblokowanego assemblera, przy czym również pokazuje się parę linijek z rzędami cyferek. Tak zachowuje się AsmOne, po tym jak skończy wykonywanie naszego kodu i odzyska kontrolę, prezentując nam aktualny stan procesora. Jeśli przyjrzymy się temu, co zostało pokazane na ekranie, to pierwsza linijka zaczyna się od „D0:” i wyświetla aktualną zawartość wszystkich rejestrów danych (w kolejności D0-D7) a druga linijka analogicznie zawartość rejestrów adresowych (A0-A7).

```

ASM-One V1.20 By T.F.A. Source » tut2.S.txt
----- ASM-One V1.20 MC680x0/MC6888x Macro Assembler (KS 1.2/1.3) -----
Original coding by Rune Gram-Madsen (1990-1991)
Additional coding by T.F.A. (1991-1992)
Additional 680x0/6888x coding by T.F.A. (1992-1993)
Release date 19-09-1993 by T.F.A.

ALLOCATE Fast/Chip/Publ/Abs>f
WORKSPACE (Max.8191) KB>100
>r
File length = 181 (=5000000B5 )
>a
Pass 1..
Pass 2..
No Errors
>j

D0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
A0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
SSP=101A7DEC USP=101A7DEC SR=0004 -- -- PL=0 -- -- PC=EOP VBR=00000000
>

```

runkowego”, bo zależnego od stanu flag statusowych w CPU) wykona skok, albo też stwierdzi, że skok nie jest potrzeby i przejdzie do kolejnej instrukcji programu.

Jeżeli BTST odczytał wysoki ('1') stan bitu odpowiedzialnego za LMB i ustawił odpowiednio flagę Z, to rozkaz BNE stwierdzi, że trzeba wykonać skok do etykiety/adresu umieszczonego w swojej drugiej części (waitmouse – zapis już bez ':'). Jeżeli natomiast BTST od-

Napisaliśmy cały kod przykładu, wciskamy Esc i wychodząc z edytora przechodzimy do linii wykonywania poleceń. Tam komendą 'a' (lub wybierając Assembler... z Menu) tłumaczymy nasz kod na język maszynowy (rozkazy zostają zamienione na ich odpowiedniki liczbowe, etykiety zaś na adresy, np. rozkaz BNE wtedy uzyskuje wiedzę, w które miejsce pamięci procesor ma skoczyć, kiedy ma ponownie wykonać pętlę). Sama asemblacja, więc to przetłumaczenie i rozłożenie naszego

Jako, że w naszym kodzie w ogóle nie używaliśmy póki, co żadnych rejestrów procesora, są one wyczyszczone przez assembler, w celu ułatwienia nam pracy. Zazwyczaj jednak nie powinniśmy ufać temu, co zostawił w rejestrach CPU jakiś poprzednio uruchomiony program i należy samemu czyścić lub wypełniać rejestry niezbędnymi nam wartościami. Do zabawy i ćwiczeń jednak taka pomoc programu jest miłym dodatkiem. Zwróćmy uwagę na 3-cią linijkę. Zaczyna się ona od „SSP”, czyli System Stack Pointer (adresu systemowego



Stosu – na razie nam nie potrzebnego do niczego), po czym widzimy wartość dla USP (User Stack Pointer). Porównajmy tą wartość (USP) z zawartością rejestru A7 (ostatnia wartość w 2giej linijce). Jak widać są takie same, bo rejestr A7 używany jest zwykle, jako wskaźnik Stosu Użytkownika i dlatego we wstępie opisując rejestry adresowe nazwaliśmy go „zarezerwowanym”. Do czego służy Stos, dlaczego dzieli się na Użytkownika i Systemowy?

W tej chwili nie jest istotne, choć polecam doczytać sobie w jakimkolwiek opisie MC680x0. Na razie zapamiętajmy, by po prostu nie używać rejestru A7, dopóki więcej nie dowiemy się o Stosie. Ostatnią rzeczą wartą zauważenia jest zapis „--Z--” (literka Z jest podświetlona). To fragment rejestru statusowego, w którym zapalony jest bit Z (flaga 'Z'ero jest „na maszcie”), a to, dlatego, że odczytaliśmy przecież wyzerowany bit stanu przycisku LMB, a skoro odczytaliśmy wartość zero – procesor i assembler odwzorowały ten stan (CPU zapalając bit Z a AsmOne prezentując nam to graficznie).

To wszystko w odcinku nr 1 – trzeba przyznać, że to spora ilość bardzo ważnego materiału. Zalecam kilkukrotne przejrzenie tego odcinka czy przeczytanie mojego wstępu i jak najpełniejsze jego zrozumienie, bo w kolejnych częściach podanych tu podstaw nie będziemy już powtarzać. Wiem, że wykrycie naciśnięcia LMB nie jest kodem marzeń początkującego kodera, ale te długie wyjaśnienia były konieczne, by móc zrozumieć kolejne części kursu. Skoro jednak przebrnęliśmy absolutne podstawy – czas zadziałać coś na ekranie.

Amiga Hardware Programming 2

Make a rasterline

W drugim odcinku kursu Photon kontynuuje kod rozpoczęty w części pierwszej. Jako, że oczekiwanie na LMB jest standardową czynnością w niemal każdym amigowym intrze czy demie, o sprawdzanie przycisku można oprzeć tzw. pętlę główną programu i to niezależnie, czy jest to jednoekranowa produkcja typu cracktro, czy trwające kilka minut pełnowymiarowe demo. We wszystkich typach produkcji najprawdopodobniej 50 razy na sekundę sprawdzany jest LMB by przejść do kolejnej części, czy też zakończyć działanie. Jako, że plany wobec kodu mamy coraz

większe, a test przycisku zyskuje na znaczenie funkcjonalne – Photon decyduje się na zmianę nazw etykiet na zgodne z ich przeznaczeniem (za co mu chwała, bo przy okazji przekazuje dobre praktyki programowania). Poprzednie, „waitmouse” staje się, więc „mainloop”, czyli pętlą główną naszego przyszłego demka. Skoro zmienił etykietę, musi też zmienić i wszystkie odwołania do niej, a więc rozkaz BNE skaczący poprzednio do początku kodu oczekiwania na LMB – teraz będzie skakał do początku głównej pętli programu.

Zaraz, przecież to, to samo miejsce... Oczywiście, że tak, co więcej, to samo miejsce może mieć kilka różnych etykiet i wszystkie zadziałają, ale już po kilku takich „objawach lenistwa” szybko pożałujemy, kiedy to wracając po paru dniach do własnego kodu zmarnujemy sporo czasu, by zrozumieć, co my tu poprzednio napisaliśmy. Doświadczony Photon od razu sam sobie ułatwia życie, robiąc krótko trwające poprawki tam gdzie trzeba. Używa do tego skrótu klawiszowego Ctrl-d, który kasuje linię, w której jest kursor, ale i też zapamiętuje ją, więc jest to wygodny sposób przenoszenia pojedynczej linii z jednego miejsca w drugie (oprócz asemlacji i skoku do wykonania – najczęściej używana funkcja AsmOne).

Wracając do pętli głównej – test przyciśniętego LMB, a więc test zakończenia wykonywania kodu dema, zazwyczaj umieszcza się na końcu, kiedy wszystkie już zaplanowane podprogramy (np. efekty wyświetlenia grafiki, scroll, sprite, plazmy, wektory, itp.) miały szansę już się wykonać, czyli kiedy cały aktualny ekran został już odrysowany. Wtedy, zanim zdecydujemy, czy będziemy opracowywać do wyświetlenia kolejną klatkę naszej zaprogramowanej „animacji”, chcemy sprawdzić, czy użytkownik zdecydował o zakończeniu wyświetlania dema, wciskając LMB. Dlatego też test przycisku stanowi pewne logiczne zamknięcie pętli głównej, a więc i naszego pomysłu na to, co chcemy aktualnie prezentować na ekranie. Oczywiście czasem wygodniej jest odwrócić kolejność i sprawdzać LMB na samym początku, ale to temat na rozważania o różnych pomysłach na ogólną budowę scenowych produkcji – do cracktr czy prostych intr, aktualnie opisywany kształt pętli głównej sprawdzi się chyba najlepiej.

Skoro mamy pętlę główną naszej produkcji - czas wymyślić treść pokazu. W tym odcinku, będzie to zmiana koloru tła na ekranie i choć może brzmieć to ba-

nalnie – wcale takie nie jest i napisanie nawet tak prostego efektu wymaga już pewnej wiedzy o hardware Amigi.

Aby zrealizować zmianę tła, nasz program musi „poczekać” na odpowiednią pozycję wyświetlania na amigowym ekranie. Teraz należałoby otworzyć sobie jakąkolwiek książkę opisującą jak wygląda tzw. organizacja ekranu w Amidze i w jaki sposób tworzy się obraz. Tu, na nasze potrzeby, odwołam się do skojarzenia z telewizorem CRT i charakterystycznymi liniami poziomymi oraz technice wyświetlania obrazu telewizyjnego, czyli legendarne „działko elektronowe” zamontowane z tyłu kineoskopu, które „strzela” w stronę ekranu i linia po linii odrysowuje zawartość wyświetlanego obrazka. Takie myślenie przyda nam się do wyjaśniania i zrealizowania wielu rzeczy, kiedy przejdziemy do tłumaczenia działania ogólnej organizacji wyświetlania efektów w demie.

Na razie przyjmijmy, że obraz wyświetlany jest od góry, linia po linii do samego dołu a każda linia od lewej strony do prawej strony ekranu. Co więcej, po narysowaniu każdej linii – następuje mała przerwa w wyświetlaniu, by działko mogło cofnąć się znów do lewej strony ekranu, przed rozpoczęciem rysowania kolejnej linii. Ta przerwa nazywa się „wygaszaniem poziomym”, bo w tym czasie działko nic nie wyświetla, więc jest „wygaszone”. Podobna sytuacja ma miejsce, kiedy po odrysowaniu najniższej linii na ekranie a przed rozpoczęciem rysowania kolejnej klatki obrazu - działko musi „powrócić” na samą górę ekranu i mamy przerwę na „wygaszanie pionowe”.

Obie powyższe sytuacje (a szczególnie ta pionowa) mają duże znaczenie dla działania wielu efektów choćby, dlatego, że można wykryć, kiedy się odbywają i dzięki nim synchronizować i sterować działaniem dema. Amiga, wyświetlając obraz PAL, w ciągu sekundy dokładnie 50 razy odrysowuje swój ekran na nowo, można więc zliczając wygaszania pionowe dokładnie określić czas wykonania efektu, można też fakt powrotu plamki rysującej obraz na górę – wykorzystać do przygotowania kolejnej klatki obrazu, uruchomić w tym czasie player muzyki, by odegrał nam kolejną porcję modułu, itp.

Takie jednokrotne odrysowanie obrazu nazywamy też „jedną ramką” (ang. one frame) i powoli zaczynamy już czuć, że nasze efekty będą szybkie i płynne, kiedy ich rysowanie zmieści się właśnie w czasie rysowania pojedynczego



ekranu. Co więcej, będziemy chcieli, by pewne rzeczy (np. scroll, wektorowa bryłka) były narysowane wcześniej, niż plamka rysująca obraz dobiegnie w miejsce, w którym ma być pokazany nasz efekt, a tu konieczne będzie sprawdzanie, w jakiej aktualnej pozycji znajduje się realizacja wyświetlania, dlatego też nawet tak prosty przykład, jak omawiany w odcinku 2 – nauczy nas bardzo przydatnej rzeczy, którą nie raz wykorzystamy w przyszłości. Tym razem naszym celem jest prosta zmiana koloru tła na konkretnej pozycji, następnym razem może to być odrysowanie cieniowanego obiektu 3D.

Skoro chcemy zmienić kolor rozpoczynając od pewnej pozycji – musimy poczekać, aż wiązka rysująca ekran od góry w tej pozycji się znajdzie. Co więcej, pierwsza linia widoczna na ekranie wcale nie jest pierwszą linią całego obrazu i to, co widzimy na ekranie monitora to trochę mniejszy fragment całości. Pierwszą widoczną linią jest linia o numerze \$2c (czyli linia nr 44!) a ostatnią \$12c (=300, czyli widocznych linii na ekranie mamy 256 – zakładając, że pracujemy w rozdzielczości 320x256). Photon zaproponował, aby zmiana koloru nastąpiła w połowie ekranu i użył bardzo pomocnego polecenia kalkulatora („? liczba/działanie”) do obliczenia właściwej pozycji. Warto chwilę przyrzeć się temu poleceniu, bo dzięki niemu, oprócz np. obliczeń – możemy łatwo konwertować liczby między systemami hex/dec/bin (sprawdźcie np. ? \$fffe, ? %10100101 i ? 127), czy pozyskiwać wartości liczbowe dla znaków ASCII (? 'A'). W poleceniu kalkulatora możemy stosować nawiasy i dzięki temu wykonywać dość skomplikowane działania, przesunięcia o pewną ilość bitów (np. ? \$ac>>1, bez spacji pomiędzy znakami), itp. Wynik prezentowany jest w postaci: wartość hex, wartość dziesiętna, reprezentacja ASCII oraz ładna rozpiska binarna, na której łatwo zobaczymy, które bity są ustawione, a które wyzerowane oraz na ilu bitach mieści się wynik. Mając wyliczony numer linii środkowej jako \$ac (dla treningu: ? (44+\$12c)>>1), możemy w końcu zrealizować nasz pomysł.

Chcąc poczekać na wybraną linię \$AC, musimy w jakiś sposób sprawdzić, jaka pozycja jest aktualnie wyświetlana. Istnieje rejestr o nazwie VHPOSR, z któ-

rego zawsze tą wartość możemy odczytać (zaglądamy do AHRM, czytamy sobie o tym rejestrze). Znajduje się on w przestrzeni adresowej, w której „widziane” przez procesor są układy specjalizowane, czyli od adresu \$dff000. Adres ten zwany jest adresem bazowym układów specjalizowanych i widząc w kodzie, że ktoś robi coś z adresem typu \$dffxxx od razu rozpoznajemy, że wykonywana operacja dotyczy naszych Custom Chips. VHPOSR znajduje się pod adresem \$dff006. W naszym programie wykorzystamy go, by sprawdzać w pętli, czy wiązka wyświetlająca obraz dotarła już do linii \$AC (stosując pętlę znaną z odcinka pierwszego: „busy wait loop”). Sprawdzenie będzie polegało na tym, że będziemy porównywać numer naszej linii – z aktualnym w rejestrze.

Dopóki będzie on inny niż oczekiwany – wykonujemy pętlę ponownie, tak jak robiliśmy to przy oczekiwaniu na LMB i rozkaz BNE będzie pracował dokładnie tak samo, czekając na bit Z. Instrukcja CMP (ang. Compare) porównuje dwie zadane liczby (oczekiwaną \$AC i zawartość pierwszego bajtu rejestru VHPOSR) ze sobą poprzez odjęcie jednej od drugiej (sprawdzamy jak działa CMP w opisie procesora), z tym, że wyniku odejmowania nie umieszcza nigdzie (poniekąd traci go), a jedynie w zależności od uzyskanej wartości ustawia flagi w rejestrze statusowym (i jeśli obie są takie same, czyli w rejestrze VHPOSR znajduje się \$AC, to wynik odejmowania daje zero i bit Z zostaje odpowiednio ustawiony). Dlatego właśnie, po porównaniu, rozkaz BNE może zdecydować o kolejnym wykonaniu pętli lub przejściu do następnej instrukcji.

Zwróćmy też uwagę, że numer linii (\$ac) to rozmiar bajtu i komenda CMP ma sprawdzić w rejestrze VHPOSR (16-bitowym!) tylko bajt spod adresu \$dff006, więc Photon zapisał ją, jako CMP.B – ze znanym nam już „określnikiem rozmiaru .B/.W/.L”. Zwracam na to uwagę, bo częstym błędem, dość trudnym do wychwycenia (komendy wyglądają zapisane poprawnie a coś działa źle) jest właśnie praca kodu na niepoprawnym rozmiarze danych.

Tu konieczne jest małe wyjaśnienie. Otóż wyświetlanie obrazu, czyli „strzelanie wiązką elektronów w kineskop”

jest wykonywane przez układy specjalizowane i niezależne od tego, co w danej chwili robi procesor. Możemy przyjąć, że oba układy (CPU i wyświetlania) pracują równolegle, dlatego procesor może czekać sobie w pętli (choć to strata czasu, bo mógłby w tym czasie liczyć np. obrót wektorowej bryły), aż oczekiwana pozycja promienia wyświetlania zostanie osiągnięta. Oczywiście wcześniej program musi odpowiednioysterować układ wyświetlania do pracy (np. tworząc czy instalując Copper-List), ale później może zająć się już innymi zadaniami a układy specjalizowane zrealizują „zlecenie”.

Skoro wyświetlanie obrazu dotarło do naszej oczekiwanej pozycji, pętla oczekująca zakończy się i możemy w końcu wykonać nasze zamierzone zadanie zmiany koloru tła. Kolor na OCS-owej Amidze reprezentowany jest przez wartość liczbową, złożoną z trzech składowych RGB (ang. Red Green Blue). Każda z tych składowych może przybierać jedną z 16 wartości (0-15) a wartości z tego przedziału mieszczą się dokładnie na 4ch bitach i najlepiej przedstawiają szesnastkowo (0 = %0000 = \$0, 15 = %1111 = \$f). Najniższa wartość dla wszystkich składowych RGB to zero – otrzymujemy więc liczbę \$000, która reprezentuje kolor czarny. Najwyższe wartości to R=15, G=15, B=15, czyli \$fff i to tworzy nam kolor biały. A co zrobić, by nasz kolor był np. zielony? Wyzerujemy składowe „przeszkadzające” (czyli czerwoną i niebieską), a wartość składowej zielonej ustawmy mniejszą, jeśli zieleń ma być ciemna (\$060), lub maksymalną możliwą, jeśli chcemy osiągnąć zieleń najjaśniejszą (\$0f0).

Tak samo możemy postąpić z innymi składowymi, co więcej, znając układ binarny RGB reprezentujący kolor, możemy „mieszać” sobie wartościami barw składowych, tak jak byśmy posługiwali się suwakami w prostym korektorze dźwięku (tony niskie-średnie-wysokie). Trzy składowe kolorów, mieszczące się na 12 bitach dają nam 4096 (0-\$fff) możliwych ustawień i stąd właśnie bierzemy się liczba kolorów dostępnych na Amidze. Kiedy i jak uzyskać je wszystkie na raz na ekranie? To już zupełnie inna sprawa, bo zazwyczaj widzimy ich od dwóch do trzydziestu dwóch i jest to związane z kolejnym aspektem har-

ASM-One V1.20 By T.F.A. Source »

>? (44+\$12c)/2

\$000000AC

172 "... " %00000000.00000000.00000000.10101100

>



dwane Amigi – dostępnymi rejestrami koloru (ang. Color Registers). Tak naprawdę, w najprostszy sposób Amiga zdolna jest wyświetlić od 2ch do 32 kolorów na ekranie, a zależy to od ilości aktualnie wykorzystywanych „płaszczyzn bitowych” (ang. bit-plane), czyli fragmentów pamięci, które układ wyświetlający traktuje jak poukładane na sobie arkusze papieru – tworzące taki warstwowy obraz. Jako, że każda z warstw to po prostu kawałek pamięci – rysując coś po niej zapalamy lub gasimy jej bity. Rysując i zapalając bity na kilku leżących na sobie warstwach możemy zauważyć, że jeśli spojrzymy na nie od góry i „w głąb” – odczytamy z nich „przeplatankę bitów”, czyli liczbę. Układy OCS zdolne są używać 5 warstw w sposób standardowy oraz dodatkowo szóstą, ale już trochę na zasadzie sztuczek sprzętowo-programowych.

Na razie skoncentrujemy się na tych podstawowych, które tworzą kombinację bitów o wartościach 0-31 i można to zapisać na 5ciu bitach (%00000-%11111). Dzięki temu powstają 32 możliwe wartości. No tak, ale skoro Amiga może wyświetlić 4096 kolorów, a 5 warstw daje tylko 32 możliwości, to znaczy, że przeplatanka bitów na warstwach nie jest kolorem, bo jak wiemy kolor zdefiniowany jest trzema 4-bitowymi składowymi R-G-B, czyli 12 bitami. Z różnych względów, twórcy Amigi wymyślili ciekawy trick, by na takiej ilości warstw wyświetlić możliwie dużą ilość kolorów – niech kombinacja bitowa z bit-plane'ów będzie numerem „schowka”, w którym umieszczona jest pełna 12-bitowa wartość koloru, a układ wyświetlający obraz każdy piksel ekranu wyświetli pobierając kolor z rejestru, o numerze wyznaczonym przez kombinację bitową na warstwach. Stąd, na 5 warstwach powstaną 32 wartości, które potraktowane zostaną, jako wyznaczniki, z którego rejestru koloru ma być pobrana pełna, 12-bitowa wartość koloru dla danego piksela. W hardware Amigi umieszczone zostały 32 rejestry-pojemniki na 12-bitowe wartości RGB, dlatego też często stosuje się paletę 32 kolorową i obowiązuje ona dla całego ekranu, chyba że... w trakcie odrysowywania ekranu poczekamy sobie na którąś linię i podmienimy daną wartość na inną.

Tak właśnie dzieje się w naszym przykładzie. Kolor tła powstaje wtedy, kiedy na żadnej warstwie, w danym pikselu, nie postawiono żadnego bitu, a więc kombinacja bitów to %00000 (= \$00), czyli zero – a takiej wartości odpowiada „Rejestr Koloru Numer 00”. Jakąkolwiek wartość RGB wpisujemy do rejestru nr

00, tam gdzie kombinacja binarna na warstwach daje zero – tam będzie wyświetlany aktualny 12-bitowy kolor z rejestru 00. Jeżeli natomiast odczekamy do połowy ekranu i wtedy podmienimy wartość RGB w rejestrze 00, to górna część ekranu będzie w kolorze pierwotnym, a drugie pół już w tym zmienionym, mimo że wcale nie przerysowywaliśmy nic na jego powierzchni – po prostu tak działa ekran i sposób wyświetlania kolorów wymyślony przez twórców Amigi. Skoro to już wiemy, zrealizujemy w końcu nasz plan, czyli instrukcją MOVE (wpisz liczbę, skopiuj wartość) prześlijmy kolor w postaci \$RGB do rejestru koloru nr 00, który znajduje się pod adresem \$dff180 (znów adres z obszaru \$dffxxx, bo wpisujemy wartość do układu specjalizowanego).

Rejestr koloru 00 używany jest, jako kolor tła całego ekranu, natomiast np. rejestr koloru 01 (znajdujący się pod adresem \$dff182, czyli 2 bajty dalej) będzie (w przypadku ekranu skonfigurowanego, jako 2-kolorowy) przechowywał wartość koloru elementów niebędących tłem, czyli narysowanych na warstwie (bity zapalone). Skoro warstw możemy mieć od 1 do 5, to i długość binarnej kombinacji będzie mówiła nam ile pierwszych rejestrów koloru będzie brana pod uwagę przy wyborze koloru. Przykładowo, dla 2ch warstw możemy mieć 4 kolory (na 2ch bitach możemy zapisać liczby 0-3) i kombinacja %00 to rejestr 00, kombinacja %01 – rejestr 01, kombinacja %10 – rejestr 02 (bo %10 = 2 decymalnie) a %11 to rejestr 03 (pamiętamy – numerujemy od zera). Zwracam uwagę na MOVE.W, bo choć wartość RGB to 12 bitów, to jednak MC68000 nie ma rozmiaru pośredniego pomiędzy bajtem (.B) a słowem (.W), wybieramy najbardziej zbliżony – sprawdzając wcześniej jednak, jaki rozmiar rejestr koloru jest w stanie przyjąć (a przyjmie nasze 16 bitów, choć użytek zrobi tylko z mniej znaczącej 12ki).

Ok, działamy dalej - zapisaliśmy nową wartość do rejestru koloru 00 (proponuję wpisać tam coś ciekawszego niż \$fff i ogólnie posprawdzać wartości dla różnych składowych RGB), zapiszmy kod do pliku (komenda „W nazwa_pliku.s” lub „Write Source” z menu) i sprawdzamy działanie programu. Zadziałało (po winno), sprawdźmy inne wartości RGB i inne numery rejestrów (\$dff182, \$dff184), po czym wysłuchajmy wyjaśnień Photona, dlaczego nie całe tło zmieniło kolor i ekran pozostaje podzielony. Dzieje się tak, dlatego że nie wyłączyliśmy jeszcze systemu operacyjnego i działamy sobie troszeczkę eksperymen-

talnie – system rozpoczynając wyświetlenia kolejnego ekranu odtwarza swoje pierwotne ustawienie koloru tła, zanim my znów w połowie ekranu je zmienimy (i tak ganiamy się z systemem w kółko, aż do wykrycia LMB). Wiedza o tym, że przy powtórnym odrysowywaniu ekranu pewne wartości wracają do swoich pierwotnych ustawień – przyda nam się gdy zaczniemy zajmować się „programem dla Coppera” czyli CopperList'ą (Copper posiada swoje niezależne od CPU własne 3 instrukcje i naprawdę dużo potrafi nimi zrobić). Na razie warto zapamiętać, że wspomniane wcześniej „wygaszanie pionowe” (Vertical Blank = VBlank = VBL) czy powrót działka elektronowego rysującej obraz z pozycji kończącej obraz (dolny-prawy róg), do pozycji początkowej (górny-lewy róg ekranu) to ważne i ciekawe z punktu widzenia koderów wydarzenia w komputerze i będziemy je wykorzystywać we właściwym czasie. Na razie kontynuujemy odcinek.

Skoro potrafimy zmienić kolor tła dla połowy ekranu, to zrobmy to dla pojedynczej linii. Można tego dokonać poprzez odczekanie, aż cała linia w nowym kolorze zostanie wyświetlona a następnie przywrócenie oryginalnej wartości koloru tła. Photon pokazuje 2 sposoby sprawdzenia, kiedy znów należy zmienić kolor. Oba są podobne, ale nie takie same. Sposób pierwszy to odczekanie na linię \$AC+1 i zmiana koloru a drugi to sprawdzanie czy linia \$AC jest nadal wyświetlana. Różnica jest mała i subtelna a to, dlatego że po wyświetleniu linii następuje powrót na lewą stronę ekranu (wspomniane wygaszanie poziome) i to o czas jego trwania różną się oba te sposoby. Czy i kiedy warto się tym zajmować aż tak dokładnie – to już temat na bardziej skomplikowane wyjaśnienia. Na razie, jako ćwiczenie, proponuję sprawdzić sobie w praktyce i zanotować oba sposoby. Zrealizujemy drugie oczekiwanie na kolejną pozycję, poznajmy sposoby zaznaczania i kopiowania bloku tekstu (Ctrl-b zaczynamy zaznaczać fragment do skopiowania, Ctrl-c to wrzucenie zaznaczonego bloku do schowka, Ctrl-f lub Ctrl-i to wstawienie bloku - odpowiednio Fill lub Insert, również dostępne z menu).

Po skopiowaniu fragmentu kodu musimy zmienić nazwy etykiet, bo assembler wyrzuci nam błąd, jeśli napotka takie same nazwy odnoszące się do różnych miejsc w kodzie. Następnie tworzymy ponownie instrukcję ustawiającą pożądaną zawartość RGB rejestru koloru 00 i sprawdzamy, czy osiągnęliśmy zamierzony efekt. W tym miejscu warto poob-



serwować, jak Photon na podstawie składowych RGB oraz swojego doświadczenia, jest w stanie odgadnąć kolor pierwotny – bez zaglądania gdziekolwiek. Zabawa z wartościami RGB w naszych prostych przykładach naprawdę ułatwi wiele rzeczy w przyszłości, a również da trochę rozluźnienia poprzez zabawę własnym kodem i pomoże nam lepiej poznać sposób tworzenia obrazu na Amidze - warto spędzić z tym trochę czasu. To praktycznie kończy zadania z odcinka 2-go.

Ostatnią rzeczą, o której wspomnę tutaj jest zdanie, kiedy Photon opisując sposób oczekiwania na zakończenie wyświetlania linii \$AC mówi, że rozwiązanie to „zmarnuje jedną linię czasu obliczeniowego”. Jak pamiętamy, nasze pętle zrobione są na zasadzie „busy wait loop”, czyli ciągłego testu pozycji promienia wyświetlającego, a skoro cały czas CPU zaangażowany jest w to sprawdzanie – nie może wykonać w tym czasie nic innego.

Jeśli chcemy, żeby nasze efekty były szybkie i płynne a przy tym ciekawe – każda zaoszczędzona ilość czasu jest dla nas cenna, bo zanim wyświetlimy kolejny obraz – musimy go przygotować, wyliczyć i odrysować. Takie czekanie jest dość niepożądane i faktycznie jest stratą czegoś (czasu), czego mamy ograniczoną ilość a „sztuką kodowania” jest wykorzystywanie tego co mamy do dyspozycji - jak najlepiej.

Amiga Hardware Programming 3

Move that rasterline

Wszystko, co do tej pory stworzyliśmy na ekranie było statyczne, a nawet w najprostszym demie – coś się musi przecież ruszać! Wprowadźmy trochę życia do maszyny.

Aktualnie nasz program czeka w pętli, aż wyświetlanie obrazu osiągnie określoną linię, która na ekranie może wystąpić przecież tylko raz. Wobec tego, po pokazaniu naszej kolorowej „kreski” (rasterline), następne jej wyświetlenie może nastąpić dopiero, gdy odrysowywany będzie kolejny ekran. Wiemy, że w systemie PAL, obraz wyświetlany jest 50 razy na sekundę. Skoro mamy już jakąś prostą synchronizację z systemem wyświetlania, to możemy ją wykorzystać, by naszą linię, co kolejną ramkę (czyli co 1/50 sekundy) przesunąć do góry lub w dół – wywołując tym samym wrażenie ruchu.

Tak właśnie robi się ze wszystkimi efektami, czy to cieniowany wektor, scroll, czy plazma. Kod programu generuje np. minimalną zmianę położenia, kształtu, obrót o jeden lub kilka stopni, przemieszczenie kolorów plazmy – 50 razy na sekundę, a oko ludzkie odczyta to, jako płynną animację.

W przykładzie z odcinka nr 2 mieliśmy stałą wartość numeru linii, na którą czekaliśmy. Jeżeli co ramkę będziemy czekali na inną, ale sąsiadującą lub w miarę bliską linię i tam umieścimy nasz kolor – widz odbierze to, jako ruch. Musimy, więc zmieniać numer oczekiwanej linii, a więc przechowywać sobie w jakiejś komórce pamięci aktualną wartość numeru linii i co ramkę numer ten zmieniać. A skoro przy omawianiu CPU rejestry opisaliśmy, jako bardzo wygodne i szybko dostępne pseudo komórki pamięci i do tego mamy jakieś nieużywane – możemy z nich skorzystać. Wykonajmy modyfikacje aktualnego kodu tak samo, jak Photon, by najpierw uzyskać efekt „niby ten sam”, co w poprzednim odcinku, ale już przygotowany do nowego zadania. Tworząc czy modyfikując swój kod, warto często wprowadzać zmiany małymi etapami (koniecznie zapisywać je na dysku, np. poleceniem 'u' od Update), sprawdzając często ich poprawność – zaoszczędzi nam to wielu rebootów i nerwów.

Warto tu jeszcze wspomnieć, że AsmOne dopuszcza pominięcie znanych nam wyznaczników rozmiarów przy rozkazie (np. pominąć .W przy MOVE) zakładając, że chodziło na o słowo - z wielu względów jednak zdecydowanie polecam dokładne dopisywanie rozmiaru. Kontynuując, warto zwrócić uwagę na to, że modyfikując kod, Photon dodaje etykietę „init:”, a za nią, ale jeszcze przed „mainloop:” - ustawia w rejestrach potrzebne wartości początkowe, by kod zadziałał tak jak chcemy. Jest to bardzo dobry i sprawdzony pomysł zarówno na program główny, jak i na mniejsze podprogramy (funkcje, procedury).

Zazwyczaj właśnie na początku następuje inicjalizacja danych roboczych (ustawienie wartości rejestrów CPU, zaprogramowanie układów specjalnych, czy odpowiednich komórek pamięci), a dopiero po tym następuje pętla główna bądź robocza, wykonująca właściwe zadanie. Czasem pętle kończy spełnienie warunku, takiego jak zliczenie odpowiedniej ilości wykonanych, dotarcie do zakładanego adresu czy ustawienie odpowiedniego bitu w rejestrze statusowym lub sprzętowym, a czasem procedura czeka na przerwanie jej np.

poprzez wciśnięcie LMB przez użytkownika. Taki układ kodu będzie często się powtarzał, warto go więc zapamiętać.

Wiemy już, że pomysłem na ożywienie poziomej linii jest przesuwanie jej w górę i w dół. Można to zrobić na kilka sposobów i do tego ćwiczenia wybrany został wybrany właściwy, bo najprostszy, ale też wcale nie najlepszy. Realizując zadanie Photon wyjaśnia, jaka jest drobna pułapka w tym podejściu, w którą możemy wpaść. Za dużo czasu nie warto nad tym spędzać, bo ten problem wynika z prostoty aktualnego przykładu. Kiedy będziemy pisać już właściwe demo, zadamy o to by zaplanowane efekty wykonywały się w odpowiednim czasie i kolejności oraz były odpowiednio zsynchronizowane z aktualną pozycją promienia wyświetlania.

W tym prostym przykładzie synchronizacja występuje trochę, jako efekt uboczny oczekiwania na pozycję ekranu (w której chcemy wyświetlić naszą linię) i kiedy zwiększymy numer kolejnej pozycji dla linii – nastąpi pewnego rodzaju nałożenie się kilkudziesięciu ustawień linii, wraz z oczekiwaniem na pozycję zwiększoną o 1 – czyli robi się bałagan. Jeżeli natomiast odejmujemy numer pozycji, czyli przemieszczamy linię w górę – nałożenie nie nastąpi i na krótką chwilę otrzymujemy to, czego oczekiwaliśmy. Gdyby jednak Photon szybko nie przerwał działania programu (klikając LMB czyli kończąc pętlę główną przykładu), linia zniknęła by na górze, po czym być może powróciła od okolic 2/3 ekranu (być może z dodatkowymi efektami ubocznymi). Rozwiązaniem tego problemu jest kilka i wrócimy do nich w dalszych częściach, tutaj wybierzemy znów najprostsze, czyli wykorzystamy kolejny raz pętlę oczekiwania na którąś z wyżej położonych linii (raster wait). Photon wyjaśnia, że czasem tego typu pętlę oczekującą lubi umieszczać gdzieś na dole ekranu, z tego powodu, że gdy ma w planie użycie Blittera do jakiejś większej pracy, zadanie to wykona się trochę szybciej, gdy obraz nie jest wyświetlany podczas przerwy wygaszania pionowego, czyli kiedy działko elektronowe powraca w lewy górny róg – dlatego jest to korzystne, wyjaśnimy sobie później, ale warto zapamiętać ten pomysł na przyszłość.

Próbując jednak ustawić pętlę tak, by oczekiwała na ostatnią linię ekranu – czeka nas niespodzianka. Otóż do tej pory posługiwaliśmy się rejestrem VHPOSR (adres: dff006) i odczytywaliśmy z niego bajt, będący aktualnie wy-



świetlanej numerem linii. Niby wszystko się zgadza – bajt może zawierać w sobie liczby od 0 (\$00 = %00000000) do 255 (\$ff = %11111111), a zazwyczaj dema czy gry pracują w rozdzielczości 320x256, więc teoretycznie w bajcie „ogarniamy” 256 linii. Problem w tym, że pierwsza widoczna na monitorze linia – wcale nie jest linią zerową, tylko (jak już wcześniej było wspomniane) \$2c (=44). Skoro pierwszą widoczną jest linia niezerowa – to czy „zabraknie” nam czegoś od dołu ekranu? Otóż nie.

Tak naprawdę „linii obrazu” (czyli łącznie tych widocznych i niewidocznych) jest aż 313, z czego górna część niewidoczna to 43 linie, a dolna (313-43-256 widocznych) 14. Niestety, na jednym bajcie, nie da się zapisać liczby 313 (\$139 = %100111001), więc potrzebny jest dodatkowy, 9ty, najbardziej znaczący bit, który znajduje się bajt wcześniej, czyli pod adresem \$dff005, w mniej znaczącym bajcie rejestru VPOSR. Kiedy jesteśmy w górnej części ekranu, bit ten jest wyzerowany i uwzględniając fakt, że wyświetlanie rozpoczyna się od linii \$2c – możemy sobie odpowiednio dobrać właściwy numer linii, jaką szukamy (N+\$2c) i spokojnie operować na mniej znaczących 8 bitach numeru. Gdy jednak przekroczyliśmy linię \$ff (a będzie to 212 widoczna linia ekranu) – wyzeruje się zawartość bajtu pod adresem \$dff006, a ustawi bit w adresie \$dff005.

Nie jest to fajne czy wygodne, ale tak zaprojektowano sposób wyświetlania

Amigi, trzeba to wiedzieć i do tego przywyknąć – czasami tak planuje się zawartość ekranu, by tej linii nie przekraczać, lub by dolna część ekranu (poniżej linii \$ff) miała inne zadanie niż 212 górnych. Photon, dla uproszczenia przykładu, właśnie się na to zdecydował, czyli ruch w dół zakończy się na linii \$f0, po czym zmieni kierunek. Wybrał \$f0, bo jest to wcześniej niż linia \$ff, a więc ta, która służy nam za synchronizację. Skoro ruchoma linia odbije się na linii \$f0, to nigdy nie dotrze do znajdującej się poniżej linii \$ff i unikniemy problemów.

Wiedząc już, dlaczego linia nie dociera do samego dołu ekranu (proponuję czytelnikom samemu rozszerzyć ten przykład by tak się stało), kontynuujemy z Photonem, realizując odbicie od górnej strony. Tu niespodzianek już nie ma, a potrzebne instrukcje najlepiej nauczyć się z video oraz sprawdzić w opisie procesora (taki opis trzeba mieć, tak samo jak AHRM – bez tego nie da się sensownie kodować, chyba że znamy już to na pamięć lub mamy gotową bazę własnych fragmentów kodu i tylko je razem składamy).

Wpisując kod przykładu, warto dawać swoje komentarze, czyli takie, które nam najwięcej wyjaśniają. Nie warto pisać rzeczy oczywistych (np. komentować, że instrukcja ADD D6,D7 dodaje rejestr D6 do D7) a raczej, że tutaj numer linii na którą będziemy czekać w danej ramce jest zwiększany lub zmniejszany o 1, w zależności od aktu-

alnego kierunku ruchu. Komentarz, jak w każdym języku, to opis lub wyjaśnienie, co dana linia lub fragment kodu ma wykonać. W AsmOne znakiem rozpoczynającym komentarz jest średnik ';' lub gwiazdka '*'. Wzorem Photona, warto sobie własny kod programu (zwany często „źródłem”, bo to miejsce z którego bierze swój początek cały program, ang. source code / source) porozdzielać komentarzami niemal „graficznie” - żeby zwiększyć przejrzystość i ułatwić szybkie poruszanie się po dłuższym programie.

Kontynuując przykład, zwróćmy uwagę na etykietę „ok1:” i skok do niej rozkazem BLO, ponieważ przeskakiwanie (omijanie) następnej do wykonania instrukcji (tutaj NEG) jest bardzo sprawnym sposobem uzależnienia wykonania czegoś w programie od spełnionego wcześniej warunku. Dlatego też, przed BLO, znajduje się komenda CMP, która jak wiemy ustawia flagi statusu, na podstawie których BLO zadecyduje – ominąć NEG, czy je wykonać.

Pamiętajmy, nasz program pracuje w pętli i podczas niektórych „obrotów pętli” warunki sprawdzane przez CMP będą takie, a podczas innych – np. odwrotne. Tutaj skok ten to rodzaj wykonania komendy znanej z języków wysokiego poziomu „IF (warunek spełniony) THEN {zrób coś}”. W naszym przypadku, można odczytać to tak: „Jeśli (warunek pasuje) to Wykonaj NEG, a jeśli (w domyśle - nie pasuje) to Omiń NEG przeskakując ją”. Polecam zapamiętać tę konstrukcję, będziemy bardzo często używać podobnych. Następnie Photon realizuje odbicie od góry na pozycji \$40, czyli nasza linia odbije się od... (\$40-\$2c = \$14)... od 20j (dziesiętnie) widocznej linii ekranu. Dalej znów (mówiłem!) stosujemy podobną konstrukcję IF jak dla BLO i mamy nasz pierwszy mikro-efekcik gotowy.

Uff, nareszcie – gratuluję dotarcia aż tutaj! Wiem, że to nic spektakularnego, ale wiedza poznana w tym artykule jest ogromna i konieczna. Dalej już będzie coraz mniej rzeczy tak podstawowych i opisowych a coraz więcej ciekawych i praktycznych. Polecam kilkakrotne przeanalizowanie tego artykułu oraz eksperymentowanie z kodem – im więcej prób i czerwonych ramek Guru teraz, tym mniej w przyszłości!

Piotr "Sachy" Sachanowicz

```
ASM-One V1.20 By T.F.A. Source » tut3.s.txt
;-----frame loop start---
    add d6,d7          ;add "1" to y position
    cmp #$f0,d7        ;bottom check
    blo ok1
    neg d6             ;change direction
ok1:
    cmp.b #$40,d7
    bhi ok2
    neg d6             ;change direction
ok2:
waitras1:
    cmp.b $dff006,d7
    bne waitras1
    move.w #$ffff,$dff180

Line: 1 Col: 1 Bytes: 548 Free:15970/1908 ----- Time: 07:03:29
>a
Pass 1..
Pass 2..
No Errors
>j

D0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
A0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 101A7DEC
SSP=101A8DE8 USP=101A7DEC SR=0004 -- -- PL=0 --Z-- PC=EOP VBR=00000000
>
```



Amiga 600

Powrót do przyszłości?

Wszystko wskazuje, że takie stwierdzenie ma rację bytu. Przed modelem 600 stoi szansa okazania się najlepszą Amigą w 30 letniej historii. Czy rzeczywiście? Na to i inne pytania odpowie Wam niniejszy tekst.

Ostatnie lata, to wzrost wszelakiej maści akcesoriów do sześć-setki. Spowodowane jest to tym, że komputer, kiedyś tak wyśmiewany, doczekał się czasów, które są dla niego odpowiednie. Jest on po prostu sexy, zarówno mały i kompaktowy, a od paru lat, kto by się tego spodziewał, łatwy do rozbudowy. Poza tym jest tani, jeszcze... Tak więc drogi czytelniku jeśli masz ochotę na 600-tkę, kupuj teraz, bo na pewno w niedalekiej przyszłości cena będzie wyższa. Powód ku temu jest tylko jeden: Vampire II, czyli karta turbo która w świecie oryginalnych Amig (od Commodore oraz Escom) zmiażdży konkurencję w zarodku. Kto wie, może jest to wielka szansa dla linii 68k na powrót z klasą? Moda na retro trwa, nie czarujmy się - Amiga nowej generacji nie przebiła się i już raczej nigdy nie przebije się do świadomości mainstreamu.

Amiga 600 to komputer, który jest następcą Amigi 500. Posiada kości ECS, mimo to nie czyni go to wykastrowaną Amigą 3000, lecz kompletnie nieudanym produktem Commodore. Miało to być budżetowe rozwiązanie dla mas. W tamtych czasach, właściwie jedyną zaletą konstrukcji była obsługa twardego dysku IDE (a to i tak nie obyło się bez wpadki z Kickstart 37.299). Niestety wadą zastosowanego złącza 44-pin była konieczność montażu dość drogich dysków 2,5 cala (porównując cenę do dysków 3,5 cala). Jednak wad było jeszcze więcej, choćby brak klawiatury numerycznej, powodujący problemy w grach czy niektórych programach. Do tego kości ECS, które dla odbiorców 600-tki, czyli graczy właściwie nie wносиły nic nowego. Zastosowany standard interfejsu PCMCIA (Type II) dopiero raczkował, a identycz-

ny procesor jak w poprzedniku, był właściwie wisienką na torcie kompletnej porażki jaką zaserwował producent swoim potencjalnym klientom.

Ponad 20 lat później sytuacja wygląda zupełnie inaczej. Niedoceniany komputer zaczął być chętnie kupowany i nadal jest sporo tańszy od Amigi 1200, nie wspominając już dużego (profesjonalnego) rodzeństwa. Myślę że właśnie ten wybór może być strzałem w przyszłościową dziesiątkę dla retro mania, który pragnie pograć w świetne gry (niestety bez tytułów AGA), popracować z Workbenchem, czy wybrać się na imprezę z takimi samymi maniakami jak on sam. Amiga jest mała, lekka, zgrabna i z odpowiednim monitorem LCD, nazwijmy to, może być niezłym "amilaptopem".

Zacznijmy więc od zakupu komputera. Oczywiście dokonamy tego na popularnych portalach aukcyjnych. Za cel postawmy sobie rozbudowanie Amigi, a w tym momencie temat konfiguracji kupowanego sprzętu można pominąć. Jedyne co jest wskazane, to próba namierzenia sprzętu w ładnym stanie, czyli możliwie białego, gdyż cały proces wybielania obudowy i klawiatury może być zarówno niełatwy, jak i pracochłonny. Temat wersji ROM (czy obsługuje twarde dyski czy też nie) jest również mało istotny. Amigę 600 i tak trzeba wyposażać się w kości 3.1, np. od firmy Cloanto, która wprowadziła poprawki m.in. w scsi.device (obsługa dużych dysków). Zależnie od budżetu oraz chęci, niezłą Amigę 600 można znaleźć w cenach od 400 do

1200 zł, ale spokojnie sprzęt do 600 zł powinien być interesującym kąskiem. Jeśli nasz zakup nie posiada wymienionych kondensatorów, warto tego dokonać. Mamy na to czas, gdyż kompletowanie potrzebnych dodatków zajmie trochę czasu. ROM 3.1 to koszty w okolicy 60-70 zł, do tego dokupienie AmigaOS 3.1, również od Cloanto to wydatek rzędu około 50 zł.

Oczywiście amigowy system pokazuje pazura dopiero na konfiguracji wyposażonej w "twardziela". Czasy się zmieniły, a więc, lepiej i wygodnie jest zainteresować się przejściówkami z serii IDE2CF czy IDE2SD. Po pierwsze jest tanio, po drugie jest stosunkowo szybko (jak na kontroler, który znajduje się na płycie głównej), po trzecie to rozwiązanie zajmuje bardzo mało przestrzeni wewnątrz komputera. Zależnie od rozwiązania ceny kształtują się od 40 do 80 zł (plus dodatkowo karta o pojemności np. 4GB).

Można też wyposażać się w dwie sztuki, celem posiadania dodatkowej karty

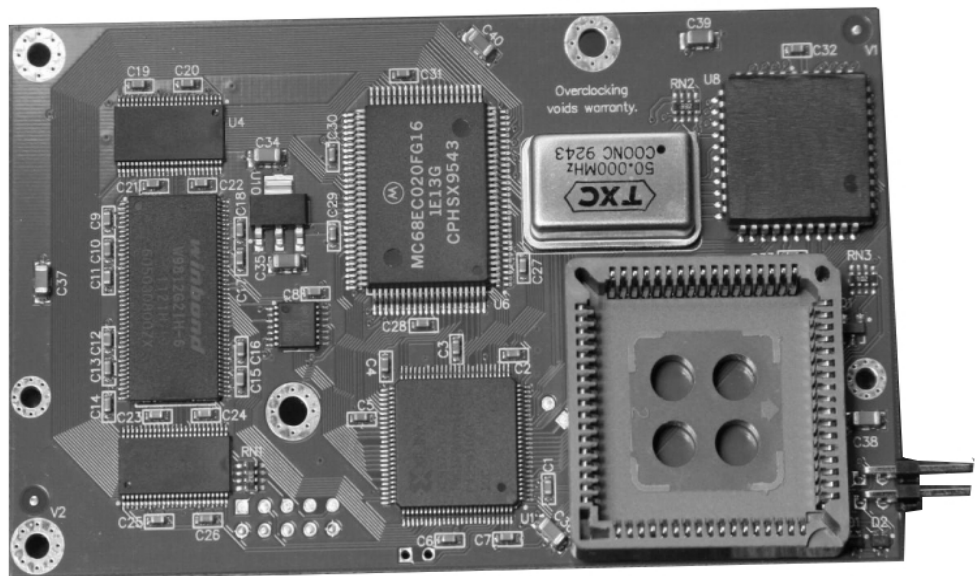


Amiga 600 na reklamie. Najszybsza w historii?



CF ułatwiającej wymianę danych. Aby to miało sens, trzeba zrobić miejsce dla karty CF np. nad slotem PCMCIA. Rozkręcanie komputera, po to tylko, aby wyjąć nośnik danych, wydaje się co najmniej mało ergonomiczne. Mimo wszystko cięcie nie naruszonej obudowy jest średnim pomysłem. Dlatego też odradzam napęd CD lub DVD - w obecnych czasach jest on zbędny. Są za to inne rozwiązania. Poniesione koszty to oczywiście początek zabawy, cała reszta będzie droższa, choć w porównaniu do innych Amig nie jest aż tak źle.

Komputer w obecnym stanie oferuje więc twardy dysk w postaci karty CF lub SD, nowszy system operacyjny 3.1 i tylko 1 MB Chip RAM. Oczywiście powyższa wartość nie jest satysfakcjonująca jako pamięć graficzna (lub operacyjna). W tym momencie przygoda zaczyna się na poważnie. Rozbudujemy Amigę 600 do stanu, w którym można podłączyć ją do monitora LCD a nawet uruchomić kultowego LightWave i pobawić się w kreowanie trójwymiarowych obrazów. Dobry początek, to zaopatrzenie się w produkt firmy Individual Computers - A604(n), oferuje on dodatkowy 1 MB pamięci Chip (czyli maksymalne 2 MB, tak jak w każdej Amidze) oraz przydatne rozszerzenia jak: moduł zegara oraz



Karta ACA620

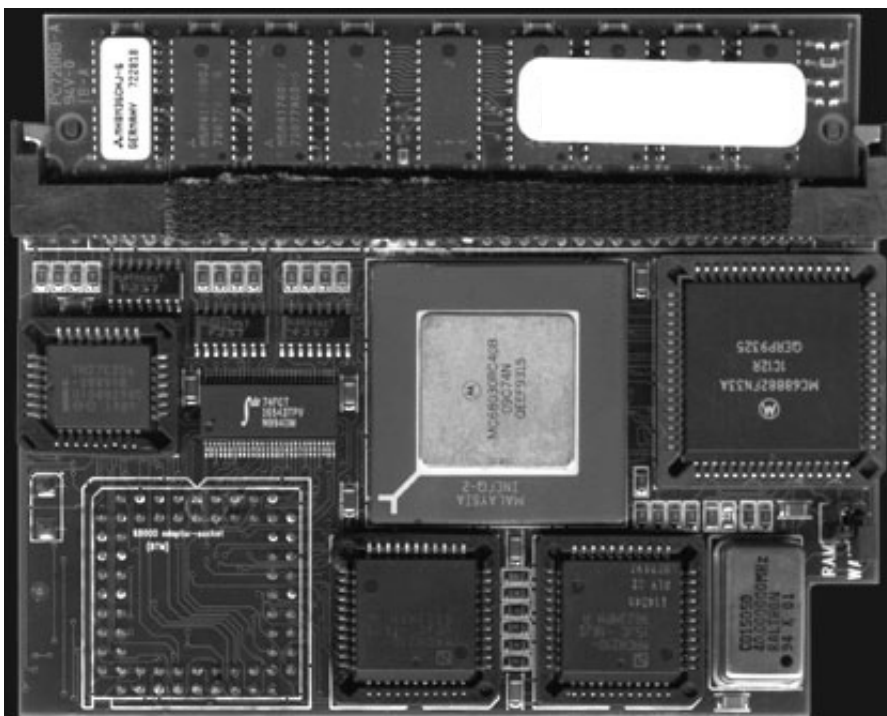
Clock Port. Poprzez ten ostatni można podłączyć inne dodatki takie jak karta USB czy karta muzyczna. Dodatkowo ten drobny kawałek krzemu wspiera Indivision ECS. Koszta które poniesiemy, to około 130 zł.

Natomiast wspomniany Indivision ECS to genialny scandoubler (podbija obraz PAL do trybów zgodnych z VGA), który poza "banalną" rzeczą jak wspieranie monitorów LCD (wraz z Scanline-Emulation), oferuje tryb Graffiti (przydatne w paru grach) a w przypadku montażu dwóch tego typu urządzeń oferuje tryb

Dual-Screen (obsługa dwóch monitorów). Nie zmienia to faktu, że kupno dwóch Indivision ECS, będzie dość trudne a zarazem kosztowne. Obecnie produkt zaczyna być ciężko dostępny w sklepach, a jego cena to około 400 zł. Bez wątpienia warto się w niego zaopatrzyć. Uzyskamy świetny sprzęt, który poza typową pracą z monitorami LCD oferuje dodatkowe rozdzielczości, takie jak 800x600 czy nawet HD720p. Jedyнным odstępstwem od zakupu powyższego urządzenia może być nabycie wspomnianego na początku Vampire II. Lecz o tym później.

Zakładając, że udało się nabyć Indivision ECS oraz posiadamy już 2 MB Chip nasz komputer zaczyna przypominać coś więcej niż zwykłą maszynkę do gier. Oczywiście Amiga nie może się obyć bez karty turbo. Obecnie, wybór jest dość spory. Niestety ciekawsze rozszerzenia można upolować tylko na rynku wtórnym. Uwaga! Będzie drogo! Jednak na początek, całkiem niezłą propozycję stanowi karta ACA-620. To kolejny produkt Individual Computers, sprzęt, który cały czas można nabyć w amigowych sklepach. Co to właściwie jest? To produkt skierowany do graczy, którzy chcą wykorzystać w pełni możliwości pakietu WHDLoad (w skrócie wspomaga on komfort obsługi i grania w gry).

Technicznie, posiadamy procesor 68EC020/16,67 MHz (czyli szybszy niż w A1200) oraz pamięć Fast RAM 11,3 MB, plus dodatkowe 2,5 MB. Oznacza to, że te dodatkowe megabajty można użyć na RAM/RAD dysk.



Karta Apollo 630



Karta standardowo uruchamia komputer z 5 MB pamięci, aby uzyskać kompatybilność z obecnie bardzo popularnymi kartami PCMCIA. Jednak dzięki oprogramowaniu ACAtune Tool można cieszyć się wspomnianymi maksymalnymi wartościami. Pieniążki jakie musimy przygotować to około 400 zł i jest to naprawdę tanio. Po przemyślanej konfiguracji systemu możemy mieć całkiem przyjemny sprzęt, który oferuje zabawę nie tylko w gry ale i umożliwi komfortową pracę z oprogramowaniem typu ProTracker, Scala czy nawet aplikacjami do grafiki 3D.

Bardzo podobną kartą, lecz sporo szybszą konstrukcją jest Furia, również oparta o procesor 68EC020, ale już 33 MHz. Oferuje do tego dodatkowo koprocessor 68882/40 MHz oraz 9,5 MB pamięci Fast. Turbo współdziała z kartami PCMCIA, lecz wtedy widziane jest tylko 4 MB Fast. Koszt to okolice

Założenie, które przyjęliśmy na początku to próba maksymalizacji najmniejszej Amigi świata. Sprawdźmy co można kupić z tzw. drugiej ręki. Oczywiście nie obędzie się bez ciągłego przeczesywania portali aukcyjnych lub dość popularnego AmiBay.com. Dżentelmeńsko, na temat kwestii finansowych nie warto rozmawiać. Obecnie ceny reguluje rynek i naprawdę trzeba być przygotowanym na kwoty rzędu do 500 Euro.

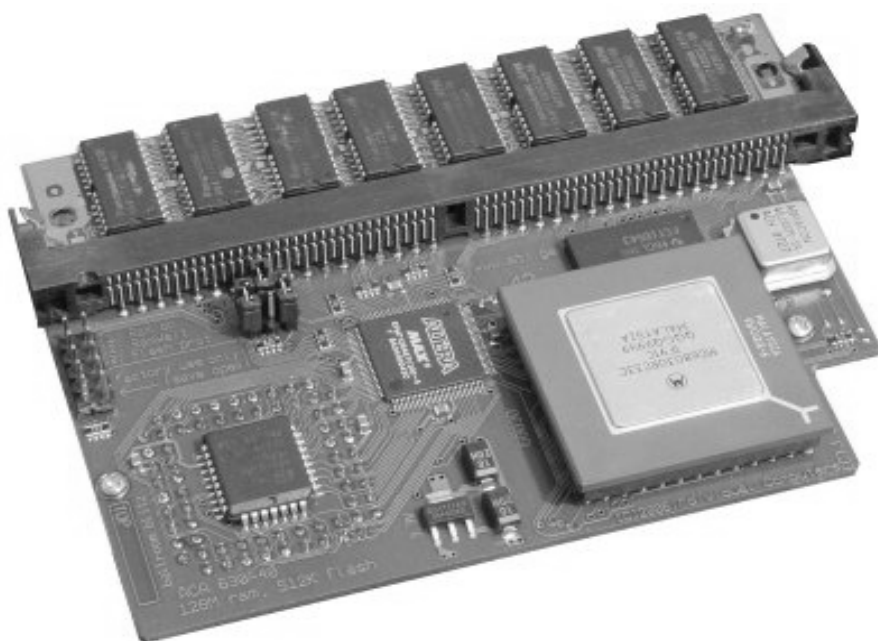
Jak dotychczas, niekwestionowanym królem dopalek dla sześć-setki jest kolejny produkt Individual Computers (tak, robi się to już nudne), o jakże porywającej nazwie ACA-630. Jest to, tak jakby starsza siostra ACA-620, jednak nie ma nawet co ich porównywać. Wersja 630 to oczywiście pełny (tzn. z układem MMU) procesor 68030, taktowany 25 lub 30 MHz, odpowiednio z 32 lub 64 MB pamięci typu Fast. Ze względu na tę ostatnią cechę, warto szukać wersji

630). Gdyby nie pojawiła się ACA-630 byłaby to najmocniejsza turbina do najmniejszej Amigi. Producent ACT Elektronik przewidział również pełną wersję procesora 68030 i koprocessora 68882 taktowanych 33 lub 50 MHz. Jak widać karta oferuje mocniejszy zegar niż ACA, jednak megaherce to nie wszystko. Jest to starsza konstrukcja, na dokładkę obsługująca maksymalnie 32 MB Fast RAM. Szkoda. Na plus należy zaliczyć to, że tu również nie ma problemów przy współpracy z rozszerzeniami umieszczonymi w porcie PCMCIA.

Powyższe karty nadają się do czegoś więcej niż przysłowiowe katowanie Super Froga. Tu spokojnie poznamy system Amigi, nawet wersja AmigaOS 3.9 nie powinna na nich sprawiać większego problemu. Będzie można zagrać w amigowego Doom, czy Gloom lub choćby Nemacs IV. Dzięki Indivision ECS będziemy mogli popracować w wygodniejszych i bardziej współczesnych rozdzielczościach w programach napisanych systemowo jak i w samym systemie. Przy przemyślanej konfiguracji, doborze odpowiedniej palety Workbenchu oraz paru dodatków można uzyskać całkiem ciekawy efekt, a oprogramowanie takie jak LightWave czy Cinema 4D pozwolą na trochę więcej niż tylko zabawę. Dodatkowo bezkonfliktowa obsługa PCMCIA, przy 32 lub 64 MB, umożliwi w miarę rozsądne korzystanie z Internetu. Oczywiście jak na możliwości i potrzeby retro.

W artykule często wspomniane jest magiczne słowo PCMCIA. W latach '90-tych, było to rozwiązanie uznane za zbyt cenne w Amidze 600 i 1200. Powodem była problematyczna dostępność rozszerzeń oraz brak dostępu do Internetu. Paradoksalnie po tylu latach, karta PCMCIA używa się właśnie głównie do sieci. Dzięki temu, dość tanio można rozszerzyć małą Amigę o port Ethernet lub nawet WiFi. Jak jest to pomocne przy przenoszeniu danych (protokół FTP) nie trzeba opisywać. Samo posiadanie możliwości i skorzystanie z легендарnej przeglądarki sieci - IBrowse na komputerze z ponad 20-letnią historią jest również bezcenną rzeczą.

Jednak PCMCIA to nie tylko Internet. To także przenoszenie danych za pomocą kart CF, podłączenie (rzadkich) napędów CD-ROM, czy nawet rozszerzenie komputera o parę drobnych i niestety wolnych megabajtów. Szczegóły oraz ofertę tego typu rozszerzeń można znaleźć oczywiście na amigowych forach, lub wpisując hasło Amiga i PCMCIA na portalach aukcyjnych.



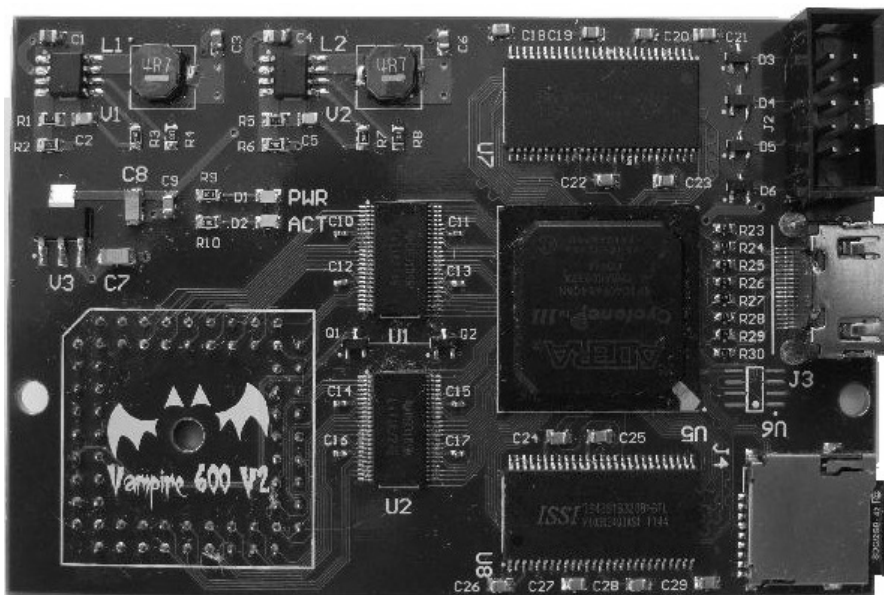
Karta ACA 630.

500 zł. Występuje też wersja bez FPU, lecz jeśli mamy okazję kupić "pełną" wersję rozszerzenia nie warto oszczędzać 100 zł.

Wspomnieć należy, że na rynku rozszerzeń do Amigi 600 mamy też polski produkt - Ninetails. Turbina jest podobna w założeniach do wcześniej wymienionych, oferuje procesor 68EC020/28 MHz i w sumie 11 MB dodatkowej pamięci. Karta współpracuje z portem PCMCIA (w trybie 4 MB Fast), jednak jej największą wadą jest praktycznie brak dostępności na rynku pierwotnym i wtórnym.

z maksymalną ilością MB. Tak wyposażona Amiga jest nawet 25% szybsza od A3000, a na dodatek kartę można podkręcić. Nie jest to godne polecenia, gdyż naprawdę obecnie ten kawałek krzemu to rarytas. Szkoda więc popsuć coś, co może być ciężkie lub niemożliwe do uratowania. Oczywiście sprzęt oferuje tak podstawowe rozszerzenia jak MapROM (kopia ROM do pamięci Fast) i współpracuje z portem PCMCIA, bez potrzeby ograniczania pamięci operacyjnej do 4 MB.

Kolejną kartą godną uwagi, lecz już mocno leciwą jest Apollo 630 (Viper



Karta Vampire II

Skoro posiadamy A600 z modulem 604n, byłoby wręcz wielkim niesmakiem nie wykorzystanie danych nam możliwości. Do komfortowej pracy modulem USB to podstawa. W sklepach cały czas dostępny jest RapidRoad USB w wersji na Clock Port (oraz dla karty sieciowej X-Surf-100), w cenie około 500 zł. Z racji wykorzystania wspomnianego portu zegara, nie można oczekiwać znacznej wydajności przy transferze danych. Jednak możliwość podłączenia pendrive i skopiowania potrzebnych plików jest bezcenna.

Co więcej, przed tak wyposażoną Amigą otwierają się szerokie możliwości, takie jak, wykorzystanie współczesnych myszek czy podłączenie karty muzycznej USB. Dodatkowo można wykorzystać przejściówki USB-Ethernet, uzyskując wolny slot PCMCIA do innych celów. Mimo ceny rozszerzenie jest to po prostu "must have" w każdej Amidze. Szczególnie, gdy chcemy wykorzystać nasz komputer we w miarę współczesny sposób. Nadmienię tylko, iż na rynku wtórnym można znaleźć podobne rozwiązanie Subway USB, jednak jeśli jest możliwość nabycia nowego sprzętu, wybór jest oczywisty.

Port zegara umożliwia też podłączenie karty dźwiękowej, takiej jak Delfina Flipper lub mającej pojawić się w przyszłości Prisma Megamix (firmy A-EON, twórcy rozwiązań sprzętowych dla Amigi NG). To świetne opcje, które umożliwią odtwarzanie plików MP3 bez obciążenia procesora. Ponadto, taka Amiga może być ewentualną bazą dla tworzenia muzyki. Pytanie pozostaje czy jednak warto tracić

możliwości jakie daje port USB? Szczególnie, że karty muzyczne w wersji na Clock Port są dostępne na rynku wtórnym, gdzie ich ceny nie należą do najatrakcyjniejszych.

Alternatywą na powyższą potrzebę, może być zakup lub samodzielna budowa MAS Player, który to podłączony pod port równoległy, umożliwi odtwarzanie plików MP3 również nie obciążając procesora przyjaciółki.

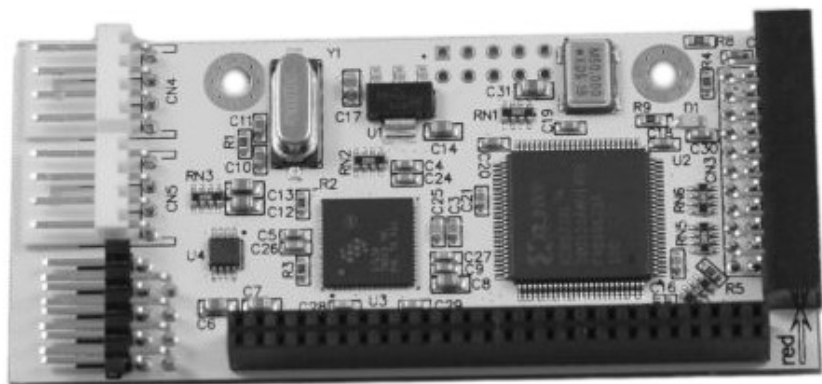
Przejdźmy jednak do rozwiązania, które w niedługim czasie rozwiąże problemy niedostatku mocy procesorów z serii 68k. Szczególnie w przypadku A600, gdzie jak widać maksymalną opcją rozbudowy pozostaje procesor 68030 i 64 MB pamięci. Mowa oczywiście o Vampire II, czyli karcie, która oparta jest o programowalny 64-bitowy procesor FPGA. Wszystko wskazuje, że będzie to ogromna rewolucja w świecie Amigi klasycznej.

Komputer, który miał być tylko następcą A500, ubogim krewnym A1200 stanie się najpotężniejszą Amigą. Wszystko dzięki nowoczesnej technologii. Okazuje się, że inwestycja w sześć-setkę może być właściwym wyborem, tym bardziej, iż w prawdopodobnie w krótkim czasie ceny tego komputera znacznie wzrosną. Dlaczego?

Koszt "wampira" kształtuje się na poziomie 150 Euro, a więc jest on bardzo atrakcyjny. Szczególnie, gdy weźmie się pod lupę parametry jakie oferuje karta. Atrakcji jest wiele, a główną z nich jest bardzo szybki procesor, na poziomie 68030 taktowanym zegarem 500 MHz oraz 128 MB w pakiecie. Mało? Do tego dochodzi jeszcze wyjście HDMI, a więc uzyskujemy tryby karty graficznej oraz wejście na kartę SD, jako dysk twardy.

Gdy jeszcze nie jesteście wystarczająco przekonani, w przyszłości dochodzą dodatki takie jak tryb audio 16-bit oraz Super AGA (czyli uruchamianie pozycji z Amigi 1200/4000). Oczywiście karta może być aktualizowana do nowszych wersji oprogramowania, co właśnie ma umożliwić wprowadzenie dodatkowych funkcji. Nawet jeśli wszystkie obietnice nie zostaną spełnione, już na obecnym etapie (składania pre-orderów), ten kawałek hardware robi z najmniejszej Amigi małego potwora. Szybkie 68k oraz karta graficzna umożliwią, Amidze klasycznej otwarcie nowego rozdziału w swojej trzydziestoletniej historii. Co powiecie na bezproblemową obsługę tak banalnych rzeczy jak multimedia, komfortową pracę w programach muzycznych i graficznych? Oczywiście, na dobrze rozbudowanych Amigach jest to już teraz (częściowo) możliwe.

Jednak w takim przypadku, koszty potrzebnych inwestycji wykraczają poza zdrowy rozsądek większości osób. Właśnie teraz, dla zwykłego fana,



Karta RapidRoad USB



otwierają się nowe możliwości za ułamek ceny, którą inwestowało się w rozbudowę klasycznego sprzętu. Nie mówiąc już o tym, że mamy do dyspozycji nowy produkt.

Wszystko wskazuje na to, że Vampire II w połączeniu z Amigą 600 może stać się retro-komputerem marzeń. Możliwe, że wpłynie to na ilość oprogramowania dostępnego dla AmigaOS 3.x, a być może doda skrzydeł projektowi AROS Vision. Czyż powoli nie okazuje się, że komputer, który do dotychczas uważany był za brzydkie kaczątko stanie się pięknym łabędziem?

Na koniec policzmy ile musimy przeznaczyć na budowę Amigi 600. Skupmy się właściwie na dwóch możliwościach: czyli obecnie dostępnym sprzęcie oraz konfiguracji opartej na Vampire II. Będą to wartości orientacyjne, nie obejmujące potrzebnego czasu lub zaangażowania dodatkowych osób w dany projekt. A600 oparta o "tradycyjny" sprzęt, czyli sam komputer plus dodatki, takie jak: IndiECS, ACA-620, RapidRoad USB oraz gadżety w postaci kart CF, przejściówek czy kart PCMCIA. W tej konfi-

guracji trzeba liczyć się z kosztami 2000 zł. Taka Amiga świetnie sprawdzi się jako maszyna do retro gier, wykorzystując swobodnie możliwości pakietu WHDLoad. Dodatkowo, gdy chcemy zastąpić stację dysków wygodniejszym rozwiązaniem, można wybrać emulator sprzętowy w postaci np. Gotek (okolice 100-200 zł). Ponadto, na takiej Amidzie spokojnie stworzymy tekst do np. Amigazynu lub Retrokompa, pobawimy się przysłowiowym ProTrackerem, Deluxe Paintem lub przygotujemy prezentację na kultowej Scali. W przypadku potrzeby posiadania koprocatora (co raczej w grach nie jest wymagane) oraz troszkę większej mocy, można za niewielkie pieniądze zastąpić ACA-620, kartą Furia z FPU.

Popatrzmy teraz na drugą opcję, czyli budowę porządnej Amigi w małej formie. Oczywiście potrzebny do tego jest nam sam komputer i właściwie tylko Vampire II. Warto też nie zapomnieć o RapidRoad USB. Oczywiście tzw. drobnica jak karty PCMCIA czy stacja dysków Gotek mogą lub będą bardzo przydatne. Z racji, że najszybsza karta turbo działa w tym samym obszarze ad-

resowym, co scan doubler Indivision, możemy go pominąć i wykorzystać możliwość pracy w trybach RTG (karta graficzna).

Niestety jest to jednocześnie sporą wadą do retro-gamingu, gdyż skazujemy się na poszukiwania zewnętrznego scandoublera (rozwiązania te mogą mieć problemy z obsługą monitorów LCD) lub dobrania takiego monitora, który będzie w stanie obsługiwać natywne amigowe rozdzielczości. A zatem ponownie trzeba szukać po forach i aukcjach. Być może, również w niedalekiej przyszłości będzie już dostępny zewnętrzny scan doubler - AMIV z portem HDMI. Jak widać, powyższe opcje wydają się być kuszące. Podsumowując, wybór zależy od potrzeb. Nie ważne jaki jest cel budowy Amigi. Obecnie model 600 jest najtańszy i stosunkowo najłatwiejszy w rozbudowie. Czy chcemy troszkę mocniejszego sprzętu, by odświeżyć sobie stare dobre czasy, czy planujemy najsilniejszą klasyczną Amiga - sześć-setka nie ma obecnie konkurencji.

Krzysztof "Radzik" Radzikowski

AMIGA.net.pl

Przejdź na naszą stronę

OFERUJEMY:

Nową literaturę przeznaczoną dla użytkowników komputerów:

COMMODORE, AMSTRAD CPC, ZX SPECTRUM, ATARI, AMIGA

POSZUKUJEMY:

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

ZAPRASZAMY DO KONTAKTU: a2@amiga.net.pl

LOAD ERROR 2015

16 - 18 października Scena Muzyczna Gdańskiego Archipelagu Kultury Raport z Party

W ostatnich latach demoscenowym centrum Polski stał się Gdańsk. Proces ten zapoczątkowało Silly Venture - impreza dedykowana platformie Atari, której sława zaczęła zataczać coraz szersze kręgi również i poza granicami naszego kraju. "Kropkę nad i" w kwestii "gdańskiej demosupremacji" postawili Sachy i Mccnex, którzy wraz z kolegami, zorganizowali w tym mieście party dla sceny amigowej i 8-bitowej o nazwie LOAD ERROR.

LOAD ERROR to scenowa impreza, która w dniach 16 - 18 października 2015 odbyła się już po raz drugi (poprzednia edycja miała miejsce jesienią 2013). Zasadniczo odbywa się ona co dwa lata i stanowi część corocznego eventu RetroKomp, skupiającego fanów retro maszyn. Informacje o party pojawiły się już na początku roku, dlatego z dużym wyprzedzeniem można było zaplanować sobie jesienny wyjazd do miasta Neptuna. RK/LE było promowane już od wczesniej wiosny, ukazały się nawet dwie kodowane invitki (z czego jedna na prestiżowym niemieckim Revision). Za bardzo oryginalny promocyjny gadżet imprezy robiła także butelka Żubrówki. Można ją było otrzymać za kodowaną produkcję na Amigę, wystawioną w scenowym konkursie.

Nasza droga na party nie była szczególnie daleka (400 km), ale z uwagi na pechowe korki (kto by tam przewidział korki w piątek po południu :) oraz rozmaite "prace wykopaliskowe" na trasie, nieco nam się wydłużyła. Na początek w - nieco osłabionym, ale wciąż silnym

- trzyosobowym składzie Ghostown (Slayer, JazzCat i Codi) zalogowaliśmy się w naszej bazie noclegowej (zabukowanej odpowiednio wcześniej). Tradycyjnie "Villa" okazała się być willą tylko z nazwy, ale jako że nie jesteśmy specjalnie wymagający (wystarczy nam śniadanie do łóżka, pakiet SPA i butelka 16-letniej whisky na powitanie) to byliśmy zadowoleni. W innych pokojach hotelowych zdążyli się już zamontować goście z Ukrainy, ale najprawdopodobniej nie byli związani z demosceną, choć ich słabość do trunków zdawała się przeczyć tej tezie.

Gdy w końcu dotarliśmy do Sceny Muzycznej Gdańskiego Archipelagu Kultury impreza trwała sobie w najlepsze. Po dopełnieniu formalności (każdy dostał stylową książeczkę do głosowania + długopis) i przejściu przez korytarz poczuliśmy się, jak po wyjściu z wehikułu

czasu w roku 1994. Powitał nas niecodzienny widok: masa kultowych retro sprzętów i kupa siedzących przy nich ludzi. Jedni grali, inni kończyli swoje produkcje.

Wrażenie niesamowitości i podróży w czasie potęgowało dobrze wyeksponowane stoisko Adama Zalepy, na którym można było nie tylko dotknąć, ale i zakupić papierową literaturę poświęconą Amidze, w tym i specjalny, zerowy numer RetroKompa, mający premierę na tej imprezie. Sam ośrodek to wprost idealne miejsce na party. Większość scenowych imprez odbywa się w dość minimalistycznych warunkach, przypominających czasem screeny z gry Fallout lub scenografię do filmu paradokumentalnego o skażeniu w Czarnobylu. Kontrast był więc spory, bo Scena GAK to zadbany obiekt o wysokim standardzie, w którym, co niektó-



Startują konkursy, powoli zaczyna brakować miejsc na sali.

rzy, mogli się poczuć wręcz nieswojo. Doskonała infrastruktura, pomocna "załoga" i nienaganna czystość (przez praktycznie całą imprezę) robiły wrażenie. Co ważne, party nie skupiało się tylko w głównej sali. Jeśli ktoś miał dość gwaru lub głośnych koncertów, mógł bez problemu znaleźć zaciszne miejsce na rozmowy ze znajomymi w innej sali lub w przestronnym holu.

Pomimo kiepskiej, deszczowej pogody, dzięki wystawionemu na zewnątrz namiotowi, wiele osób wybierało integrację na świeżym powietrzu. Znajdowało się tam stanowisko z całkiem niezłym piwem (z lokalnego browaru Amber) i - już nieco gorszymi, acz wciąż zjadliwymi - hamburgerami (zgodnie z obowiązującymi trendami nie zabrakło też wegetariańskiej wersji rzeczony buły). Piątek upłynął w miłej atmosferze na rozmowach, wymianie poglądów i sączeniu płynów. Kolega Sordan z narażeniem życia dbał o to, aby żadna pozostawiona na stole butelka, nie wpadła w niepowołane ręce, choć niestety w jednym wypadku mu się to nie udało. Na party cały czas coś się działo, ilość atrakcji i punktów programu była spora. Odbyło się nawet typowo scenowe seminarium pod tytułem "Gru-

jemna, o tyle w praktyce nie było już tak różowo, szczególnie w nocy i podczas deszczu. Na szczęście w drodze do/z GAK pomagały nam - całkiem tanie - taksówki, a w dzień i przy dobrej pogodzie kompas, busola i organoleptyczna analiza mchu na mijanych drzewach i kamieniach. Po "intensywnym piątku", który zakończył się dla nas o 3 nad ranem wybraliśmy - rzecz jasna -

była dość niepokojąca. Obstawialiśmy, że bierze nas za zbiegłych pacjentów zakładu psychiatrycznego. Tymczasem na party place czekał już na nas nasz stary, dobry znajomy - Noname/Haujobb, który to specjalnie na tą okazję dotarł do Gdańska samolotem prosto z Niemiec. Warto zaznaczyć, że była to jego pierwsza wizyta na polskim party (i jednocześnie pierwsza w Polsce).



Trwa 8-Bit Pixel Graphics Compo.



Nastroje wśród publiki dopisywały.

pa TROPYX, podsumowanie 20-lecia działalności na scenie c64", podczas którego V-12 ciekawie opowiedział o ostatnich dwóch dekadach działalności tej formacji. Na LOAD ERROR nie sposób było się nudzić.

Niestety, o ile trasa z naszej "villi" na partyplace, oglądana wcześniej na Google Maps, wydawała się łatwa i przy-

opcję Taxi, gdyż nikt nie miał już sił na analizę mchu na drzewach. W sobotę rano, dzięki pożywnemu śniadaniu, które zaserwowano nam w pensjonacie, nabraliśmy sił, uzupełniliśmy elektrolity i już około 12 byliśmy gotowi na kolejny partyjny dzień. Swoją drogą, mina Pani w stołówce, która przygotowywała nam śniadanie i przysłuchiwała się naszym porannym demoscenowym rozmowom,

W plecaku miał cenny ładunek, laptopa - z jeszcze ciepłymi - źródłami najnowszej wersji naszego demo. Od paru tygodni pracowaliśmy bowiem wspólnie nad produkcją, które mieliśmy zamiar wystawić wieczorem w Amiga Demo Compo. Tradycyjnie demo nie było jeszcze ukończone, więc czekała nas dość pracowita sobota. Na szczęście demo było już w mocno zaawansowanym stadium, więc do dopracowania zostały szczegóły. Czyli zupełnie inaczej niż przy kilku naszych wcześniejszych produkcjach.

Losy takiego „Human Traffic” ważyły się praktycznie do końca; demo oddawaliśmy 5 minut przed rozpoczęciem konkursu, modląc się w duchu do Wisznu/Sziwy/Jahwe, żeby tylko się nie zawiesiło na compo machine. Nic dziwnego zatem, że byliśmy wyluzowani. Wokół naszych komputerów stały omszałe fiaszki, słychać było chłopot zacnych cieczy oraz pełne niewymuszonej erudycji, polsko-niemieckie dialogi w rodzaju: „Was is das?”, „Das ist Kwas”.

W międzyczasie w sobotę odbył się także konkurs na najlepszą grę w ramach RetroKomp GameDev Compo 2015. Co prawda przy produkcji kilku tytułów pracowali również scenowcy, jednak więcej szczegółów na temat tego konkursu, jak samej części Retro-

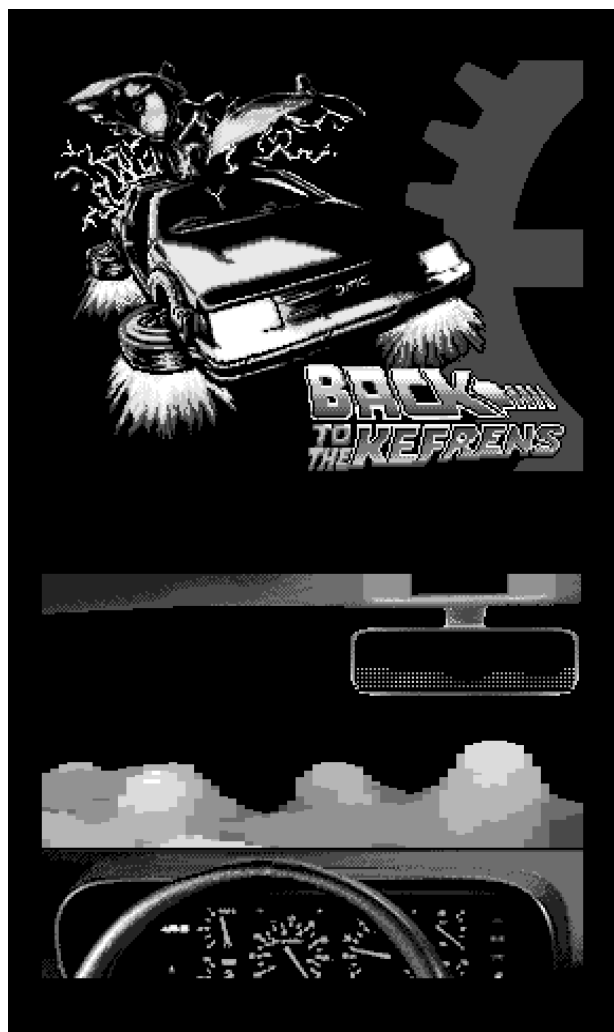
Komp imprezy, przeczytacie w tekście, który napisał Twardy.

Gdy nareszcie skończyliśmy demo i przetestowaliśmy je na maszynie konkursowej, z poczuciem dobrze spełnionego obowiązku wyskoczyliśmy na miasto coś przekąsić. Po kilkunastu minutach błądzenia znaleźliśmy knajpę z tradycyjnym polskim jadłem, która spełniała dwie żelazne zasady Magdy Gessler: było w niej sporo ludzi (znaczy dobre) i jej menu było krótkie (znaczy świeże). Noname nie ukrywał zachwytu polskimi specjałami: żurkiem, pierogami i lokalnymi piwami. No tak, nie ma co się dziwić, wszak w Niemczech jedzą tylko te swoje okropne wursty z grilla.

Kolejne godziny upływały na rozmowach i integracji z partyzantami. Tutaj pozdrawiamy szczególnie wesołą ekipę z Chełma. Ich pozytywna aura, jest chyba najlepszą rekomendacją organizowanego przez nich AmiParty. Trzeba się będzie tam w końcu kiedyś wybrać. Warto zaznaczyć, że z głośników przez



Grafika "When The World Explodes", autor: Slayer / Ghostown. Pierwsze miejsce w 8-Bit Pixel Compo.



Screensy z demo "Back To The Past" grupy speccy.pl. Pierwsze miejsce w 8-Bit Demo Compo.

cały czas sączyły się cichutko przyjemne, spokojne moduły. Na bigscreenach (bo były aż dwa!) pojawiały się natomiast demo i scenowe grafiki, a nie żadne "wątpliwie zabawne" filmy z YouTube, co zdarzało się często na innych tego typu imprezach. Bardzo dobry pomysł! Partyzanci bawili się świetnie, czasami może nawet aż "za świetnie". Co prawda, nic złego się nie stało, ale momentami włos jeżył się na głowie, gdy, widział się jakiegoś rozbawionego uczestnika, niepewnym krokiem balansującego ze szklanką, kilka cali od sprzętu wartego grube tysiące.

Pod wieczór rozpoczął się blok konkursowy. Ilość wystawionych prac, jak na realia polskiego party, była imponująca. Konkursy podzielone były na dwie kategorie główne (8 bit i Amiga) i odpowiednie podkategorie (muzyka, grafika, demo). Co ważne, wszystkie produkcje

(z małymi wyjątkami) odtwarzane były z prawdziwych "koszernych" sprzętów. Był to nie lada wyczyn logistyczny, ale sprawnie i płynnie zmieniać compo-maszyny.

Podczas wszystkich konkursów, bez większych pauz, prace pokazywały Amiga 1200, Amiga 500, SAM Coupé (na party "spotkały się" trzy egzemplarze tych rzadkich komputerów, co było najprawdopodobniej najliczniejszym ich „meetingiem” w historii polskich parties), ZX Spectrum, c64 (w dwóch wersjach SID), małe Atari, Amstrad CPC i PC. Szacunek Panowie! Atmosfera panująca w sali przypominała z jednej strony stare dobre polskie parties z lat 90-tych, a z drugiej profesjonalny klimat panujący na Revision.

W 8-bit Pixel Gfx Compo zaprezentowało się 9 prac, z czego aż siedem namalowanych zostało w trybach graficznych c64. Większość stanowiły prace klimatyczne i nastrojowe, oparte o ciekawe pomysły, choć nie zabrakło, rzecz jasna, również i banalnych scenowych motywów pokroju kobiecej (?) fizjonomii. Najlepszą grafikę na "komodorka" zaprezentował węgierski grafik Leon. Jego "Lost Romantic" (c64 Hi-Res) to malarska, nieco melancholijna i niepokojąca wizja wesołego miasteczka. Intrygujące wrażenie robiła też grafika Darklighta "Close encounter", będąca jego pierwszą próbą zmierzenia się z - bardzo trudnymi - trybami graficznymi ZX Spectrum. Z kolei jeden z niżej podpisanych (Slayer) zmierzył się po raz pierwszy z trybami SAMa

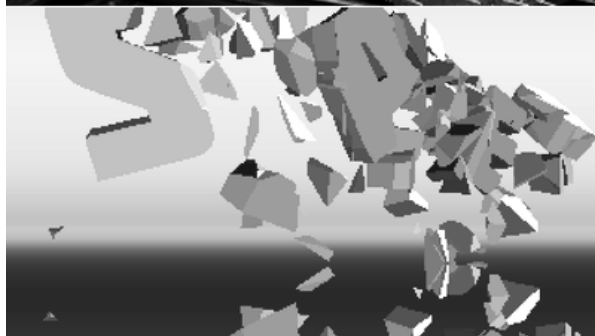
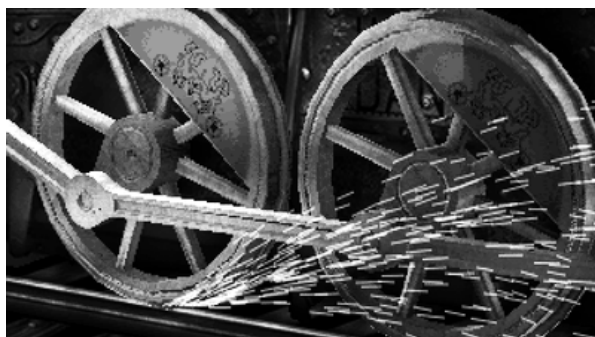
Coupé. Trzeba przyznać, że maszynka ta ma całkiem spory, dość niespotykany, w świecie maszyn 8-bitowych potencjał graficzny. Dziwi więc fakt, że ilość ręcznie pikselowanych rysunków ((nie licząc konwersji) stworzonych na SAMa Coupé - w całej jego scenowej historii - jest śladowa. Doprawdy, zaskakujące.

W 8-bit Demo Compo wystawiono 7 dem: cztery na Commodore 64 i po jednym na ZX Spectrum, małe Atari i platformę SAM Coupé. Z produkcji na c64 wyróżniało się trackmo "Secora", wydane z okazji 20 lecia grupy Tropyx oraz demo Sin(64) (które zrealizowali Jaro i Cane). Ciekawie zaprezentowały się pozostałe oldschoole platformy. Honor Atari uratowała (nie pierwszy raz zresztą) niezłomna grupa Lamers prezentując "State of The Lane". To króciutkie demko, będące parodią kultowego trackma Spaceballs wywołało bardzo żywiołową reakcję publiczności, a jeden z kolegów wymagał nawet błyskawicznie przeprowadzonej resuscytacji krążeniowo-oddechowej, bo zakrztusił się sokiem z rabarbaru. Ucieszna parodia oryginalnego soundtracku jest w stanie rozbawić każdego, nawet jeśli jest akurat w trakcie leczenia kanałowego bez znieczulenia. Lamersi po raz kolejny pokazali, że jeśli chodzi o demoscenowe poczucie humoru, są po prostu bezkonkurencyjni.

Swoją drogą zadziwiająco niska produktywność na LE'15 sceny spod znaku Atari była dość częstym tematem rozmów na party. Dziwiło to tym bardziej, że w 2015 nie odbyło się przecież atarowe Silly Venture. "Dynx" na ZX Spectrum 48 + AY to z kolei niewielki one- (a właściwie two-) screener. Intro zostało niemal w całości napisane na party, podczas prawie 24-godzinnego maratonu. Tygrys po raz kolejny udowodnił, że ma cojones, oddając produkcję godzinę przed konkursem. Jednak największy aplauz zebrało demo, pokazane jako ostatnie. Komputer SAM Coupé nigdy nie był jakoś specjalnie popularny na scenie, nawet w czasach swojej świetności.

Panowie z grupy specy.pl pokazali jednak, że jest to doskonała maszyna do scenowania. "Back To The Past" to ukłon w stronę znanej filmowej serii. Wszak to właśnie 21 października 2015

w drugiej części trylogii, Marty McFly wylądował wehikułem czasu w przyszłości, natomiast pierwsza część filmu ukazała się 30 lat wcześniej. Demo (które miało ukazać się na Forever 2015) jest bardzo dopracowane, a



Screeny z dema "Last Train To Danzig" grup Ghostown & Haujobb. Pierwsze miejsce w Amiga Demo Compo.

wszystkie jego poszczególne części składowe są na naprawdę wysokim, światowym poziomie.

Wild Compo to taki konkurs z cyklu "mydło z kapustą" tudzież "groch z podwielem". Trafiły do niego wszelkie produkcje, które nie pasowały do żadnej innej kategorii. Ponad połowa prac była

utworami muzycznymi, które nie mieściły się w regulaminie Oldskool lub Amiga Music Compo (np. były w formacie mp3). Pozostała część stanowiła na szczęście znacznie bardziej przekrojowy materiał. "Kombi-nacja" grupy Vulture Design to solidny music disk na C-64 z coverami starego kultowego zespołu Kombi, a "Party vs Rodzina" grupy Obrazzers to zabawne video demo okraszone cytatami ze scenowych forów, na popularny ostatnio temat, dręczący wielu scenowców wybierających się na party. Również i my wystawiliśmy coś w tym kompiecie, "Trybuna Loda" to video demo w którym pokazaliśmy jak wiele łączy demoscenowy świat z polityką i showbiznesem. Sądząc po reakcjach zgromadzonych, chyba nam się udało.

Konkursy muzyczne były dwa. Amiga music i Chip. O ile w pierwszym przypadku większych wątpliwości nie było – wystawiać można było tylko moduły oparte na samplach i skomponowane na dowolnym trackerze z Amigi klasycznej, o tyle zasady drugiego konkursu wymagają drobnego wyjaśnienia. Niby są proste. W muzyce nie ma być sampli tylko syntezowane dźwięki generowane przez układ dźwiękowy. Ale co w takim razie z AHX, który przecież nie generuje brzmień w czasie rzeczywistym, a tak się składa, że jest najpopularniejszym programem do robienia chipów na Amigę? Czy powinien być wykluczony z chip compo? Na szczęście podejście organizatorów do tematu okazało się zdrowe i zamiast tworzyć pedantyczne podziały uznali, że utwory w formacie AHX to również chip music. Chociaż może nie do końca kośzerny. Podobny stosunek do tematu mają zresztą orgowie Revision czy Riverwash. Druga ciekawa rzecz to brak mp3 compo. Ma to swój sens, bo o ile do zrobienia porządnego modułu potrzeba typowo scenowych umiejętności, o tyle już mp3 może zrobić ktoś niemający o scenie pojęcia. LOAD ERROR nie jest zresztą wyjątkiem pod tym względem - na szwedzkim Datastorm było tak samo.

W Chip Compo startowało 16 kawałków na różne platformy: c64, Amiga, Atari ST, Atari XE/XL, ZX Spectrum, a nawet

Amstrad CPC. Czyli multiplatformowość pełną gębą. Utwory były puszczone w czasie rzeczywistym z prawdziwych maszyn, więc łatwo sobie wyobrazić ilość roboty związanej z przepinaniem kabli czy regulacją poziomów...

A mimo to konkurs przebiegł sprawnie. Wszystkie moduły udało się odtworzyć, żadna maszyna się nie zawiesiła i, co najważniejsze, w trakcie tych cokolwiek ryzykownych i nerwowych zabaw z prądem nikt nie zamienił się w ogień św. Elma (a przynajmniej nic takiego nie zauważyliśmy). Kilka utworów wybijało się ponad przeciętność. Na przykład energetyczny, melodyjny numer „Fourth of a Kind” autorstwa XTD, który najwyraźniej wraca do dawnej formy. Albo „Courage” 505 ze świetnymi arpeggiami i znakomitymi solówkami. „Reload” AceMana to z kolei rozbudzany, zadziorny kawałek z doskonale wyważonymi proporcjami między melodyjnością i tanecznym pulsem. Warto też odnotować udział w compo legendarnego, powracającego na scenę Rocka/Finish Gold.

W Amiga Music konkurowały ze sobą kawałki oparte na samplach stworzone na rozmaitych trackerach. Podobnie jak na Revision, nie było podziału na multichannele i 4-channele. Łącznie w compo startowało 9 modułów, w tym 4 w formacie Protrackera, 3 w formacie DigiBooster 2, natomiast 2 stworzone zostały na DigiBoosterze 1. Stylistyka i sam poziom były zróżnicowane. Drum'n'bass, chińskie ska, house, a nawet cokolwiek krotchwilny dance. Było kilka ewidentnych żartów, kilka eksperymentów i kilka normalnych mo-



Siłowanie na rękę? Nie! To V0yager odbiera nagrodę za 1sze miejsce w Amiga Music Compo.

dułów. Kawałek Voyagera prezentował się najefektowniej ze wszystkich, bo napisany został w zawrotnej prędkości F01. Dzięki temu patterny przelatywały po ekranie niczym uchodźcy przez niemiecką granicę! Ale muzycznie ciekawszy był na przykład „Sky Garden” Micha. Atmosferyczny, a jednocześnie energetyczny d'n'b z ładnymi, rozmarzonymi melodiami pianina i dobrze brzmiącymi instrumentami.

Tradycyjnie już w trakcie kompotów część publiczności tańczyła. Było to swoiste połączenie tańca św. Witta z danse macabre wykonane z wdziękiem i lekkością rusalek leśnych.

Amiga Graphics Compo stało na wysokim poziomie. Zgłoszono 9 prac, z czego większość spokojnie mogła by ozdobić takie imprezy jak Gerp czy Revision. Po wielu latach stagnacji, to już drugie party w tym roku w Polsce (po wrocławskim Decrunchu), gdzie odbyło się Amiga Gfx Compo z prawdziwego zdarzenia.

Podobnie jak w 8-bit Pixel Gfx Compo grafiki były w większości nastrojowe. „Wielki las” autorstwa Aszu to monumentalne, ciekawe retro roboty, nieco podobne w klimacie do pracy Koyota. Jego „Seventh Daughter” to świetna, bardzo dopracowana praca w amigowym hiresie z masą szczegółów i detali. Brawo za pomysł i nowoczesne podejście do pixel artu. Równie psychodeliczna paleta kolorów charakteryzuje „The Fountain” Darklighta. Jego „krajobraz we wnętrzu głowy” to tradycyjnie solidny rysunek, dobry detal i niezły pomysł. „Teapot Genie” Carriona to niespodzianka w konkursie. Ładnie zaprezentowała się ta praca na bigscreenie, bo jest kolorowa i poprawnie skomponowana. Świetny amigowy debiut, oby tak dalej. W „Riverside” (autorstwa PikkuMyy - nowej graficzki na pixel-art'owej scenie) głównym „napędem” są klimat i atmosfera.



Oczekiwanie na ogłoszenie wyników i wręczenie nagród.



Grafika "Death Is A Lonely Business", autor: Slayer / Ghostown. Pierwsze miejsce w Amiga Gfx Compo.

Dobra kolorystyka i kompozycja kompensują nieco brak detali i wykończenia i praca ta równie świetnie wypadła na dużym ekranie. Warto też wspomnieć o kolejnych dwóch pixel-art'owych debiutach (Codi i Miro).

W końcu nadszedł czas na główny punkt programu czyli Amiga Demo Compo. Od wielu lat, na polskich parties, ilość amigowych produkcji można było policzyć na palcach lewej ręki drwala cierpiącego na Parkinsona. Nie jest tajemnicą, że amigowa scena w Polsce przeżywa kryzys, dlatego wiele osób zastanawiało się, jak będzie tym razem. Publiczność, która szczerze wypełniła salę, czekała jednak nie lada niespodzianka. Oto, po raz pierwszy w Polsce od roku 2003 miała ona szansę obejrzeć "pełnokrwisty" konkurs na amigowe demo. Do konkursu zgłoszono aż 8 produkcji.

Jako pierwsze zaprezentowane zostały te mniejsze z nich. 40kb intro, "Hello-DemoScene" grupy Outsiders, działające na Amidze OCS, to raczej typowa pchółka, ale trzeba podkreślić, że jest to jednocześnie amigowy debiut, nowego kodera, Flopsa. Pierwsze scenowe lody przełamane, mamy nadzieję, że niedługo pojawią się kolejne produkcje. "Roball" to niewielkie, ale bardzo zgrabne cracktro (również na Amigę 500), które zakodował Asman. Intro zawiera ładne logo i przyjemna muzycz-

kę. "Bumpy" to z kolei ciekawa 4 - kilówka na duże Amigi, autorstwa Phibrizzo, a "RSE20" to intro na OCS, świętujące urodziny grupy Resistance z kodem Sachego. Intro zawiera świetną scenę z "kosmicznymi antenami" oraz zacny motyw przesuwających się

chmur. „Teapot Genie” to natomiast amigowy debiut Carriona, znanego komodorowskiego grafika. Demo zawiera raczej proste efekty, ale podane na talerzu smacznej grafiki, designu i doprawione przyjemną, nieabsorbującą muzyką. Kolejna całkiem niespodziewana rzecz. Czekamy na więcej!

Kolejna produkcja - „Black Light” - kooperacja grup Wanted Team i Lamers to już rzecz cięższego kalibru. To próba przedstawienia natury tytułowego światła czarnego, a więc światła o mniejszej długości fali niż światło widzialne. Jak pamiętamy z fizyki, światło ma naturę dwoistą, dlatego niektóre efekty (promieniujący, spiczasty, szescian, „żyletki”) obrazują jego postać korpuskularną, a inne (znakomity błyszczący plazmowy robal) – falową. Ciekawe i oryginalne. Atutem dema jest spójny, minimalistyczny design przeniknięty chłodem ultrafioletu. Na uwagę zasługują bardzo współczesne i rzadko widziane na OCS glitche i dobra synchronizacja z zimnym soundtrackiem electro.

Ciekawostką jest fakt, że demo chodzi również na A1000 (z rozszerzoną pamięcią ChipRam do 512k)! Należy dodać, że demo stworzyli Sachy i Mccnex - główni organizatorzy party (z graficzną pomocą at0ma/Lamers), co wydaje się wyczynem z gatunku "mission impossible". Ponoć pracują już nad książką o



Screen z video dema "Trybuna Loda" grupy Ghostown. Pierwsze miejsce w Wild Compo.

robotycznym tytule "Jak zorganizować party, napisać na nie dwa dema i nie dostać hemoroidów?".

Spore poruszenie wywołało demo „Wildcat”. To pierwsza produkcja KK (znanego z PC i Atari) na Amigę. Chyba nikt się nie spodziewał tego dema, a już na pewno nikt się nie spodziewał, że będzie tak dobre! Zaskakująco szybki i płynny wolfenstein, niezły teksturowany

mista, obok Noname'a, był Hellfire - utalentowany maestro kodu, udzielający się dotychczas bardziej na scenie pecetowej. Produkcja powstała nieco przypadkowo, jako swoisty test "zdolności koalicyjnej" pomiędzy grupami Ghostown i Haujobb, sprawdzenie nowych narzędzi i środowiska opracowywanego demosystemu. Z zakładanej miniaturki, projekt nieco nam się jednak rozrósł (choć wciąż trwa tylko niespełna dwie minuty). "Last Train..." przeznaczona jest na Amigi z układami AGA, procesorem 060 i 32 MB pamięci fast i zawiera kilka nawiązań do samego party i Gdańska.

Zgodnie z tytułem, motywem przewodnim jest (steampunkowy, a jakże!) pociąg, a najbardziej zaawansowanym efektem, który

Żubrówki, które organizatorzy wręczyli twórcom za kodowane produkcje. Niecodziennego klimatu (tydzień po party miały się w Polsce odbyć wybory parlamentarne) dodawała też urna wyborcza, do której wrzucało się karty (a raczej książeczki) do głosowania. Noc upłynęła pod znakiem integracji, dyskusji i dźwięków generowanych przez Skippa. Około trzeciej nad ranem trzeba było jednak już wracać. My do hotelu, a Noname na swój powrotny samolot. W tym miejscu serdeczne podziękowania należą się Flopsowi, który, jako jeden z nielicznych trzeźwych uczestników party, wykonał swoim auto - ścigaczem szybki kurs na gdańskie lotnisko.

W niedzielę rano, w drżących rękach nieco wymęczonych już partyzantów, królowały głównie zimne napoje i wody mineralne. Ogłoszenie wyników i wręczenie nagród przebiegło w sympatycznej atmosferze i uroczystej oprawie. Były zdjęcia, gratulacje, wywiady, wiwatujące tłumy i rzucane na scenę części damskiej i męskiej bielizny :)



V-12 podsumowuje 20-lecie działalności grupy TROPYX.

sześcian obracany w osi X, świetny efekt przypominający ruchomy spektrogram... Naprawdę sporo jak na debiutanta. Oczywiście każde pierwsze demo cierpi na niedoróbki i podobnie jest i tutaj: brak przejść między efektami, grafika jest ascetyczna, kolory niespójne i miejscami po prostu brzydkie... Demo powstawało jednak w pośpiechu, w dodatku bez udziału grafika, więc można przymknąć na to oko.

Konkurs zamknęło "Last Train to Danzig", wspomniane już wyżej demo, nad którym pracowaliśmy wraz z dwójką koderów z grupy Haujobb. Drugim progra-

można zobaczyć, jest rozbijanie trójwymiarowych brył za pomocą fragmentacji Voronoi. Największy nacisk położyliśmy na klimat, synchronizację i dynamikę całej prezentacji. Amiga Demo Compo zakończyło konkursowe zmagania, ale impreza trwała dalej, w czym pomagały butelki



Amiga Demo Compo. Na bigscreenie produkcja grupy Resistance.

Warto zaznaczyć, że na LOAD ERROR oprócz nagród rzeczowych (np. kuponów promocyjnych na zakup wydawnictw Adama Zalepy, płyt DVD i różnych amigowych gadżetów) zdobyć można było - jak za starych czasów - także (a może przede wszystkim) "żywą gotówkę". Nagrody w wysokości 700 zł za miejsce pierwsze w konkursie na kodowaną produkcję, czy 300 zł w indywidualnym to przyzwoite kwoty, mogące stanowić dla zwycięzców swoisty "zwrot kosztów" za wejściówkę, pobyt w hotelu, dojazd na party czy wyżywienie. Jest to z pewnością miły, symboliczny akcent dla najlepszych twórców za włożoną pracę. Po ceremonii na bigscreenie zaprezentowano jeszcze film o demoscenie: The Art of the Algorithms i ludzie pomału zaczęli rozchodzić się w różne strony świata.

Dla nas było to bezsprzecznie najlepsze polskie party w ostatnich latach. Po raz pierwszy od 2003 roku udało się zorganizować amigowe demo compo. W dodatku z niesamowitą jak na polskie warunki ilością i jakością prac. I z czterema prodkami zupełnych debiutantów! Chyba nikt się czegoś takiego nie spodziewał. Okazało się również po czasie, że niewiele brakowało, a ilość amigowych dem mogła śmiało przekroczyć 10, gdyby nie kilka pechowych zdarzeń. Praktycznie jedynym minusem imprezy był brak informacji o rodzaju produkcji w trakcie demo compo. Widzowie nie mieli pojęcia, czy to demo, intro, czy może 4-kilówka. Ale organizatorzy zdają sobie z tego sprawę i do dzisiaj czują się winni, aplikując sobie co tydzień godziną dawkę samobiczowania. Panujący na party klimat był niesamowity, a organizacja wyborna. Trzymamy kciuki, aby w 2017 roku odbyła się następna edycja i wypadła co najmniej równie dobrze oraz oby ożywienie na polskiej amiscenie okazało się trwałe. RK/LE Team - dziękujemy.

Slayer & JazzCat

Autor zdjęć: Artur Makowski



LOAD ERROR 2015 WYNIKI

Amiga Demo Compo

- | | |
|--|------------|
| 1. "Last Train To Danzig" - Ghostown & Haujobb | (620 pkt.) |
| 2. "Wildcat" - DMA | (540 pkt.) |
| 3. "Black Light" - Wanted Team & Lamers | (527 pkt.) |

Amiga GFX Compo

- | | |
|---|------------|
| 1. "Death Is A Lonely Business" - Slayer / Ghostown | (507 pkt.) |
| 2. "Riverside" - PikkuMyy / Dreamweb | (468 pkt.) |
| 3. "Teapot Genie" - Carrion | (429 pkt.) |

Amiga Music Compo

- | | |
|---|------------|
| 1. "Modulo-3" - V0yager / Nah-Kolor | (425 pkt.) |
| 2. "Sushi Ska" - JazzCat / Ghostown | (404 pkt.) |
| 3. "Her Dick In Ur Mouth" - AceMan / Appendix | (397 pkt.) |

8-Bit Demo Compo

- | | |
|-----------------------------------|------------|
| 1. "Back To The Past" - Speccy.pl | (620 pkt.) |
| 2. "State Of The Lame" - Lamers | (478 pkt.) |
| 3. "Secora" - Tropyx | (417 pkt.) |

8-Bit Pixel Compo

- | | |
|---|------------|
| 1. "When The World Explodes" - Slayer / Ghostown | (588 pkt.) |
| 2. "Makrosmatik" - Odyn / Tropyx ^ Vulture Design | (399 pkt.) |
| 3. "Night" - Data / Tropyx | (398 pkt.) |

Chip Music: Compo

- | | |
|---|------------|
| 1. "Fourth Of A Kind" - XTD / Lamers | (513 pkt.) |
| 2. "Hello Blocky People" - JazzCat / Ghostown | (485 pkt.) |
| 3. "Reload" - AceMan / Appendix | (467 pkt.) |

Wild Compo

- | | |
|-----------------------------------|------------|
| 1. "Trybuna Loda" - Ghostown | (599 pkt.) |
| 2. "Kombi-nacja" - Vulture Design | (434 pkt.) |
| 3. "Party vs Rodzina" - Obrazzers | (424 pkt.) |

**Lista obejmuje trzy pierwsze miejsca
w poszczególnych konkursach.**

LOAD ERROR 2015

GameDev COMPETITION

O tym, że klimaty retro komputerowe przeżywają swoisty comeback nie trzeba nikogo przekonywać. Materialnymi przykładami tego trendu są chociażby wysokie ceny, które osiągają stare komputery na aukcjach internetowych, coraz większe przebijanie się tematu do mediów mainstreamowych czy duża ilość tematycznych imprez, które organizowane są zarówno w Polsce, jak i za granicą. Fakt, że do czytelników trafiło niniejsze czasopismo, też jest dowodem na to, że stare komputery mają się dobrze i nie zostały zepchnięte na margines historii, tudzież do składu elektrośmieci.

Z mojego punktu widzenia, sam fakt, że paru przeżywających kryzys wieku średniego facetów wydobyło z piwnic i strychów swoje Atari, Amigi i Spectrums, żeby przeżyć coś na kształt powrotu do lat młodzieńczych nie jest żadnym osiągnięciem. Dla mnie, zawsze cechą charakterystyczną środowiska skupio-

nego wokół tych komputerów była jego kreatywność. Tworzenie produkcji scenowych, gier, nowych systemów operacyjnych, prowadzenie portali, organizacja zlotów i imprez czy redagowanie magazynów zawsze stanowiły dla mnie wartość nadrzędną wobec przysłówowego klikania w Cannon Fodder i kłótni na forach dyskusyjnych.

Organizowana już po raz czwarty gdańska impreza RetroKomp (oraz po raz drugi dołączające do niej party scenowe Load Error) jest wydarzeniem, podczas którego spotykają się pasjonaci tworzący wartość dodaną dla całego środowiska. Budynek Gdańskiego Archipelagu Kultury stał się miejscem corocznych spotkań ludzi zaangażowanych na demoscenie, twórców gier i oprogramowania użytkowego, osób aktywnych na portalach internetowych czy autorów publikacji na temat retro sprzętu. W roku 2013 podczas drugiej edycji imprezy przeprowadzony został konkurs dla twórców gier, w którym wystawiono 11 produkcji na platformy takie jak Amiga, Commodore 64, Atari 8-Bit i ZX Spectrum. Jako pomysłodawca konkursu, chciałem przede wszystkim zachęcić ludzi do tworzenia czegoś więcej, niż tylko setek postów na forach internetowych. Konkurs okazał się du-

żym sukcesem, więc postanowiliśmy, że w nieco odświeżonej formie zostanie on powtórzony także w czasie edycji 2015.

Wzorem 2013 roku, podczas imprezy RetroKomp/Load Error 2015 odbył się ponownie zorganizowany przeze mnie konkurs RK Gamedev Compo, w którym swoje produkcje wystawiali osoby tworzące gry na nasz ulubiony, stary hardware. Regulamin konkursu przewidywał, że przyjmowane są produkcje na wszystkie możliwe modele starych komputerów i konsol, oczywiście z wyłączeniem PC i nowoczesnych konsol. W odróżnieniu od edycji 2013, produkcje nie mogły być przygotowane w game-makerach, czyli programach do tworzenia gier, warunkiem dopuszczającym do konkursu był samodzielnie napisany kod i oryginalna oprawa. Wyjątkiem od tej reguły były porty gier z innych platform, o ile tytuł wcześniej nie był dostępny na platformie docelowej. Do udziału w konkursie dopuszczone były zarówno ukończone gry, jak i wersje demo i preview. W głosowaniu na najlepszą grę mógł wziąć udział każdy z uczestników scenowej części RK/LE 2015, votka do głosowania znajdowała się w książeczce, którą goście otrzymywali razem z biletem na party.

Ostatecznie do konkursu zgłoszono aż 16 produkcji na 7 różnych platform sprzętowych, co było znacznie lepszym wynikiem, niż w 2013, kiedy to do konkursu stanęło 11 gier. W RK Gamedev Compo przewidziano nagrody rzeczowe za miejsca 1-3, a z okazji obchodzonej w 2015 trzydziestej rocznicy premiery komputera Amiga, przewidziano też dodatkowe nagrody dla dwóch najwyżej sklasyfikowanych gier na tę platformę sprzętową.

Nagrody ufundowane były przez nieocenionego Lotharka, za pierwsze miejsce można było otrzymać komputer MIST, a za kolejne miejsca na „pudle”, emulatory stacji dysków HXC. Dodatkowe nagrody amigowe zostały zasponsorowane przez Sanjyuubi i Ar-



Nadzieja na nową grę RPG wyładowała na miejscu 8.

chitect1200 z forum PPA i były to rozszerzenia pamięci Chip i Fast do Amigi 600 oraz dedykowany do Amigi 600 i 1200 adapter CF2IDE. W ostatniej chwili dodatkowe fanty w postaci Plipboxa do Amigi i Raspberry Pi w obudowie stylizowanej na Amigę 3000 podrzucił nam Jenot z Chełm Amiga Legion.

Sam konkurs odbył się w formie prezentacji, ze względów praktycznych gry były pokazywane w formie nagranych wcześniej materiałów wideo oraz zrzutów ekranowych. Kolejność prezentacji gier była alfabetyczna. Pełen zapis przebiegu konkursu można obejrzeć na YouTube, pod poniższym adresem:

https://www.youtube.com/watch?v=bysogV6eX_o

Ogłoszenie wyników i wręczenie nagród, odbyło się następnego dnia rano.

W dalszej części relacji znajdują się opisy każdej z gier wystawionych w compo.

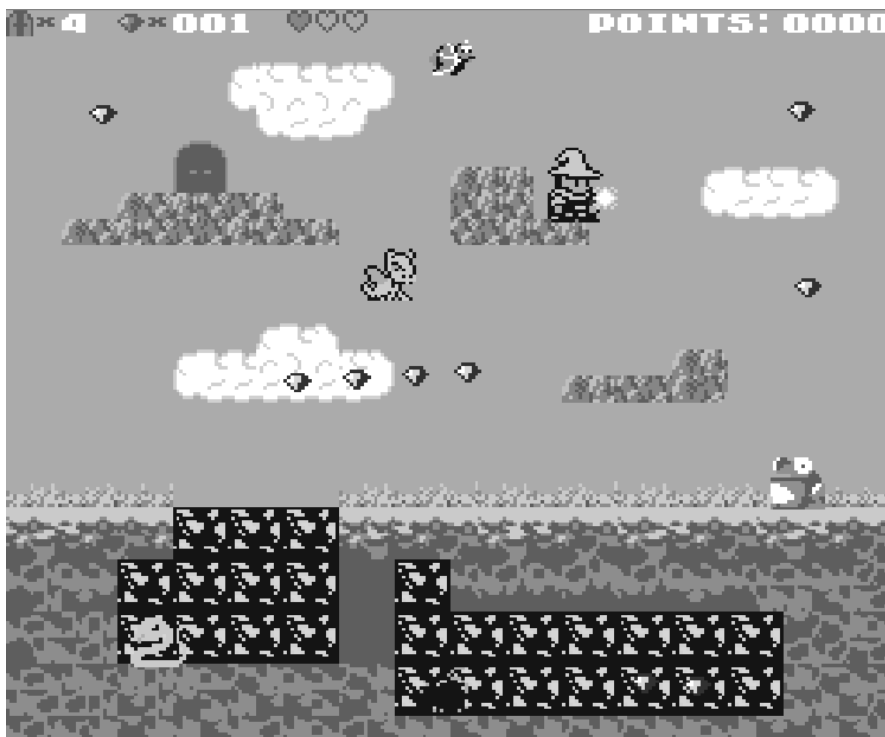
**1. Miejsce:
The Wizard, Story Unknown
Nintendo Entertainment System
(270 punktów)**

Pierwsze miejsce w konkursie Retro Gamedev Compo zajęła gra platformowa The Wizard, na 8-bitową konsolę NES. Autorem kodu gry w języku C, a także oprawy graficznej jest Axi0maT, czyli Piotr Król z redakcji portalu:

retroage.net

Muzykę dostarczył znany scenowy muzyk, AceMan z grupy Appendix. Gra The Wizard jest platformówką, w której pokonujemy kolejne, jednoekranowe etapy, a naszym zadaniem jest zebranie odpowiedniej liczby diamentów i dojście do wyjścia. Dużą zaletą gry jest bardzo dobra oprawa audiowizualna. Wrażenia estetyczne są podobne, jak przy produkcjach ze „złotej ery” konsoli NES.

Gra powstawała praktycznie do ostatniej chwili przed konkursem, muzyka AceMan'a pojawiła się w niej dosłownie rzutem na taśmę i całkiem możliwe, że to właśnie skoczny chiptune zdolnego, scenowego muzyka ostatecznie zdecydował o wygranej w konkursie. Zaprezentowana wersja składała się z 7 poziomów i nie jest jeszcze grą ukoń-



Zwycięska gra, czyli czarodziej w akcji.

czoną. Axi0maT cały czas dopieszcza swoje dzieło, a jego celem jest fizyczne wydanie gry na cartridge. Myślę, że pierwsze miejsce w konkursie było jak najbardziej zasłużone, The Wizard stanowi dobry przykład, że nowe gry mogą powstawać także na pozornie zamkniętej platformy retrokonsolowej. Dodatkowe punkty należą się także za dopracowaną oprawę i wysoką grywalność.

Myślę, że po dodaniu ostatnich szlifów i wydaniu gry w formie fizycznej, The Wizard stanie się przebojem. Na chwilę obecną gra jest do pobrania w wersji na emulatorze Nintendo.

Link:

http://www.retroage.net/pliki/The_Wizard_1.00.zip



2. Miejsce: Tanks Furry Amiga OCS (240 punktów)

Tanks Furry był przez wiele osób z amigowego forum PPA typowany na mro-
wanego faworyta do pierwszego
miejsca w RK Gamedev Compo. Osta-
tecznie gra zajęła drugie miejsce, jed-
nak przyznać należy, że konkurencja
była duża, a wersja gry pokazana
w konkursie była jeszcze daleka od
ukończenia. Należy też pamiętać, że
w subkategorii amigowej, Tanks Furry
zajęło miejsce pierwsze.

Nad grą pracował zespół w składzie:
Paweł „Juen” Nowak (kod w ASM),
Krzysztof „Koyot 1222” Matys (grafika),
oraz Marcin „Doomer” Święch (efekty
dźwiękowe). Tanks Furry, jest rozwinię-
ciem koncepcji gry Tanks, znanej
z konsoli NES, rozgrywka polega na
kierowaniu czołgiem i eksterminacji
komputerowych, bądź sterowanych
przez innych graczy przeciwników.
Wersja konkursowa gry prezentowała
się bardzo dobrze pod względem grafiki
i grywalności, autorzy udostępnili kilka
przykładowych leveli, jednak brakowało
jeszcze chociażby muzyki, za którą
w ostatecznej wersji gry odpowiada Si-
mone „JMD” Bernacchia oraz trybu
multiplayer.

Na moment pisania tego artykułu gra
jest już ukończona, posiada oprawę
muzyczną oraz możliwość rozgrywki do



Czołgi zawsze w modzie - przynajmniej na Amigę.

4 graczy jednocześnie. Na wygłodnia-
łych Amigowców czekać będzie 40 mi-
sji, a produkcję będzie można pobrać
za darmo. Przewidziana jest także ko-
mercyjna wersja pudełkowa, która po-
winna być już dostępna do nabycia na
stronie sklepu:

amiga.net.pl

w momencie, kiedy będziecie czytać te
słowa. Wersja konkursowa dostępna
jest tutaj:

**[http://www.po-
uet.net/prod.php?which=665
00](http://www.po-
uet.net/prod.php?which=665
00)**

3. Miejsce: Quarrior Atari XL/XE (215 punktów)

Gra Quarrior „przyleciała” na RK Ga-
medev Compo aż ze Słowacji. Za kod
gry w assemblerze odpowiada Matosimi,
który do spółki z PG stworzył też grafikę
gry, natomiast autorem muzyki jest Po-
ison.

Quarrior działa na Atari XL/XE, wyma-
ga 64 KB pamięci RAM, wykorzystuje
modyfikację stereo. Produkcja jest grą
logiczną, w której zadaniem gracza jest
ustawienie w linii, co najmniej 3 kloc-
ków w tym samym kolorze. Podobnie
jak w słynnym Tetris, z czasem zwięk-
sza się stopień trudności gry.



Nieśmiertelny Tetris w zupełnie nowej odsłonie.



Z jednej strony mamy tutaj dość klasyczny, żeby nie powiedzieć często powielany pomysł, ale z drugiej strony, jak widać gracze nadal nie mają dość takich gier, co w połączeniu z dobrą oprawą audiowizualną, dało Quarrioron wysokie, trzecie miejsce.

Link:

<http://www.pouet.net/prod.php?which=66537>



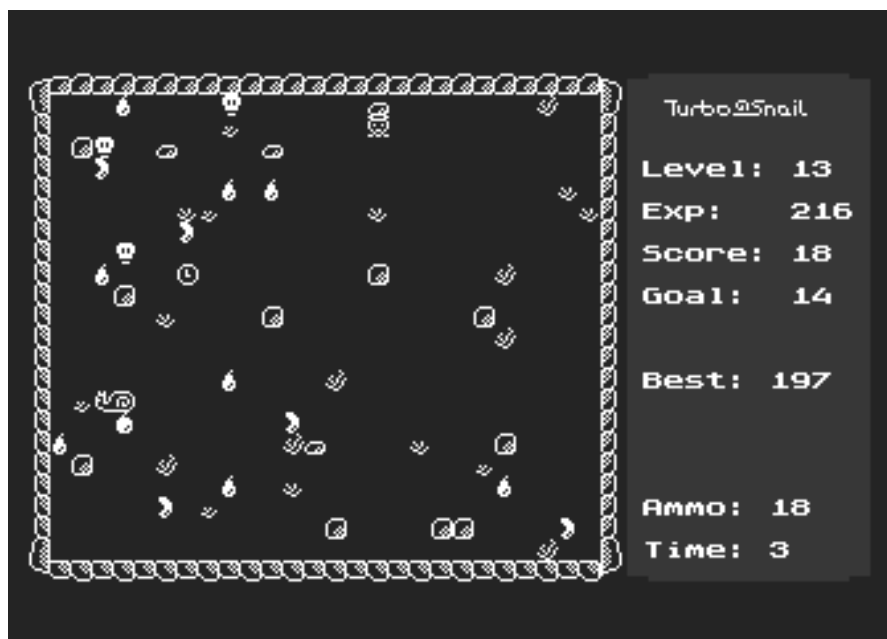
4. Miejsce: Turbo Snail Atari XL/XE (207 punktów)

Tuż za Quarrioron, uplasowała się kolejna gra na „małe” Atari, przygotowana przez grupę anticSHOP, w postaci Wojtka „Bocianu” Bocińskiego, który napisał kod w języku C, i przygotował oprawę dźwiękową oraz Krzysztofa „Kris3d” Kuca, pomysłodawcę gry i autora jej grafiki. Gra działa na małym Atari z przynajmniej 48 KB pamięci. Bohaterem Turbo Snail jest ślimak, którego zadaniem jest zebranie z kolejnych plansz „dobrych przedmiotów”, przy jednoczesnym unikaniu tych „złych”. Żeby nie było tak łatwo, gracz musi w ograniczonym czasie zdobyć określoną liczbę punktów SCORE, określonych dla każdego etapu, jako GOAL.

Gra ma dość ascetyczną, ale czytelną i estetyczną oprawę graficzną, na ekranach tytułowych przygrywa nam przyjemny chiptune, natomiast rozgrywce towarzyszą efekty dźwiękowe. Dużym plusem jest możliwość zapoznania się z pełnym opisem gry i instrukcją, która wyświetla się po planszy początkowej. Autorzy Turbo Snail zapowiadają kolejne tytuły, więc mam nadzieję, że 4. miejsce na RK Gamedev Compo, jeszcze bardziej zachęci ich do kreatywności.

Link:

<http://www.pouet.net/prod.php?which=66536>



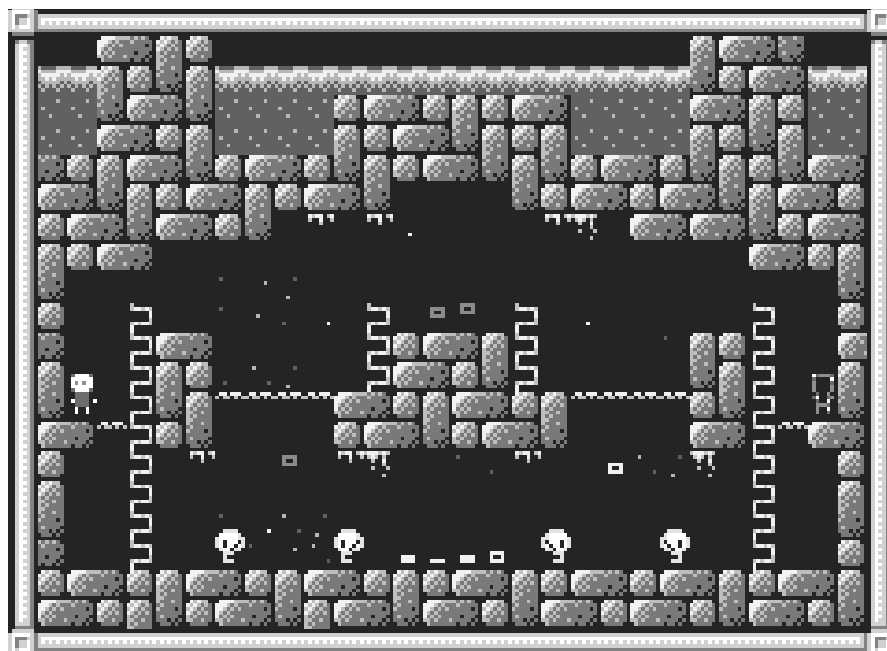
Prosta grafika może być atutem.

5. Miejsce: Alter Ego Amiga ECS (196 punktów)

Drugą najwyżej sklasyfikowaną produkcją amigową okazało się Alter Ego, czyli port gry pierwotnie napisanej na konsolę NES. Została ona przeportowana przez Asmana z grupy Wanted Team, który uzyskał na to zgodę autora gry na NES, Shiru, a także pomysłodawcy pierwowzoru gry na ZX Spectrum, Denisa Gracheva. Alter Ego ma w sobie

elementy zarówno zręcznościowe, jak i logiczne. Bohater, którym kierujemy jest jakby w dwóch postaciach. Poza główną, poruszamy przezroczystym duchem, który znajduje się na przeciwnym rogu planszy i wykonuje względem głównej postaci ruchy, będące jej lustrzanym odbiciem.

Zadaniem gracza jest zebranie wszystkich artefaktów z kolejnych plansz, przy jednoczesnym unikaniu przeciwników. Trudność polega też na tym, że w niektóre miejsca może dotrzeć tylko nasz „duch”. Gra nie należy do łatwych i wy-



O dwóch takich... w labiryncie.

maga sporo praktyki, żeby pokonywać kolejne plansze. Alter Ego ma bardzo przyjemną oprawę graficzną i proste efekty dźwiękowe. Do zagrania wystarczy dowolna Amiga z 512 KB pamięci RAM.

Link:

http://www.ppa.pl/artykuly/download/ftp/alterego-_adf.zip

6. Miejsce: Tronowo Commodore PET, Commodore 64 (194 punkty)

Bartek „TomRain” Poletajew przygotował grę, która zwróciła na siebie uwagę niecodzienną platformą sprzętową na której działa, niezbyt często bowiem powstają gry na Commodore PET. Zapewne dlatego jest ona napisana w Basicu i uruchamiana w trybie tekstowym. Tronowo oparte jest na znanym pomysłe gry Tron, gdzie kierujemy pojazdem zostawiającym za sobą długi barwny ślad, którego gracz nie może przeciąć. Zasady gry są podobne do popularnego węża.

Tronowo ma też tryb multiplayer, w którym dwóch graczy stacza pojedynek na ekranie, nie tylko unikając własnego śladu, ale także zwalczając się nawzajem. Ciężko mówić o oprawie graficznej, czy dźwiękowej, gra działa, bowiem w czystym trybie tekstowym 80x25. Bartek przygotował też port gry na Commodore 64.

Wersja PET:

<http://csdb.dk/getinter-nalfile.php/142713/pet-tronowo.d64>

Wersja C64:

<http://csdb.dk/getinter-nalfile.php/142712/tronowo64.d64>

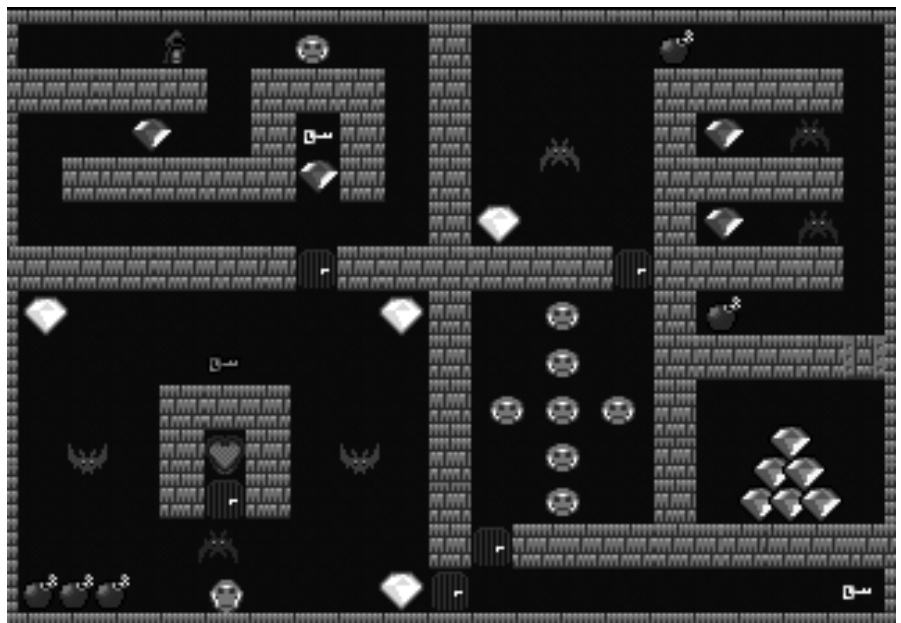
7. Miejsce: Crazy Priest Amiga ECS (185 punktów)

Crazy Priest to napisana w Amosie gra zręcznościowa, będąca połączeniem Dynablastera z Pac-Manem i Key Masterem (gra pokazana na RK Gamedev Compo w 2013). Kierujemy postacią kapłana przemierzającego kolejne plansze labiryntu, na których musi zbierać diamenty, zabijać przeciwników oraz zbierać klucze otwierające kolejne części etapu. Nasz kapłan może strzelać

do przeciwników i podobnie jak w Dynablaster - podkładać bomby. Gra nie należy do najłatwiejszych, nie tylko ze względu na szybko poruszających się przeciwników, ale także z uwagi na ograniczoną liczbę dostępnych dla gracza bomb i amunicji.

Uwagę zwraca dobra oprawa audiowizualna, grafika zarówno w menu, jak i w samej grze jest ładna i kolorowa, na ekranach tytułowych możemy posłuchać niezłego modułu, natomiast rozgrywce towarzyszą różnorodne sample. Autorem kodu gry jest Sordan, za grafikę odpowiada Aszu, a za dźwięk JMD.

Gra działa na kościach ECS i procesorze 68000 oraz wymaga 2 MB pamięci Chip, jednak na takiej konfiguracji jest bardzo wolna. Docelowo więc zalecany jest FAST RAM i szybszy procesor.



W AMOSie też można napisać ciekawą grę.



Jedyna gra w konkursie dla Commodore PET.

Link:

<http://www.po-uet.net/prod.php?which=66516>

**8. Miejsce:
RPG Tech Demo
Amiga ECS
(181 punktów)**

Demo engine przyszłej gry RPG dostarczone rzutem na taśmę przez Selura. Tak naprawdę trudno napisać coś więcej, bo do dyspozycji było tylko jedno „pomieszczenie”, po którym gracz mógł sobie pochodzić. Silnik przypomina Dungeon Mastera i graficznie wygląda bardzo ładnie, jednak przed Selurem jeszcze dużo pracy i mam nadzieję, że ostatecznie powstanie z tego silnika ciekawa gra RPG. Na razie 8. miejsce na zachętę.

Link:

<http://amiga.net.pl/dlf/rk/rpgtechdemo.adf>

**9. Miejsce:
Laserball 2015
Atari ST
(178 punktów)**

Gra logiczna na Atari ST, przygotowana przez Niemca, Thomasa Ilg. Nieco zaskoczyła mnie niska pozycja tej gry w konkursie, ponieważ jest to pełna



W konkursie brała udział gra logiczna z Niemiec.

i kompletna gra, którą autor (poza udostępnieniem opcji darmowego ściągnięcia) wydał także w formie pudełkowej. Celem Laserball jest doprowadzenie promienia lasera z punktu A do punktu B z wykorzystaniem innego dla każdej planszy systemu luster, które obracamy i przemieszczamy tak, aby promień trafił w wyznaczone miejsce.

Gra ma bardzo dobrze wykonaną grafikę, okraszoną wstawkami w stylu japońskiej mangi, poziomy są zaprojektowane czytelnie, a wszystkie ich elementy pieczołowicie wypiksowane. Oprawa dźwiękowa w postaci chiptune także nie odstaje od pozytywnego obrazu całości.

Poziom trudności gry jest bardzo zróżnicowany, mało doświadczeni gracze mogą zacząć od niewielkich i nieskomplikowanych układów z wyłączonym ograniczeniem czasowym, a dla ekspertów autor przygotował masywne poziomy, których przejście zajmie graczowi kilkanaście minut. Upływający czas możemy przedłużać zbierając apteczki, zwiększające zasób pozostałych minut.

Laserball 2015 nie jest zupełnie nową produkcją, jest to rozbudowana wersja gry, którą Thomas Ilg stworzył jeszcze w roku 1991 jako nastolatek.

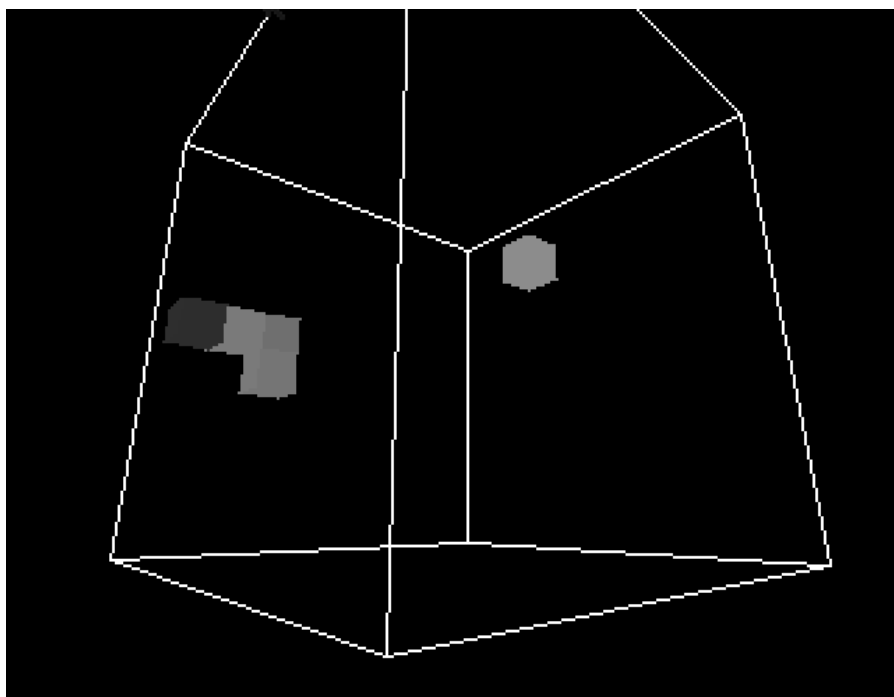
Link:

<http://www.hd-video-film.com/laserball/>

**9. Miejsce:
Ziggy
Amiga AGA
(178 punktów)**

Dziewiąte miejsce konkursu przyniosło remis, ponieważ zarówno opisany powyżej Laserball 2015, jak i amigowy Ziggy zdobyły po 178 punktów. Ziggy, zakodowana przez znanego z forum PPA Roberta „Phibrizzo” Krajcarza, jest to próba implementacji pomysłu znanej gry Snake, w środowisku 3D.

Zaprezentowana na RK Gamedev Compo wersja, to dopiero wczesne preview, bez oprawy dźwiękowej i z ubogim menu, jednak kompletne na tyle, żeby można było zapoznać się z koncepcją autora, która okazała się bardzo ciekawa.



Trójwymiarowy świat na Amigę 1200.

Tak jak w standardowym wężu musimy zjadać porozrzucone tu i ówdzie klocki i uważać, żeby wając się po planszy, nie dotknąć „własnego” ciała, jednak w grze Ziggy cała zabawa odbywa się w środowisku 3D. Nasz wąż porusza się nie po płaskiej planszy, lecz wewnątrz zawieszonego w przestrzeni sześcianu i to w pełnych trzech wymiarach. Zabawa jest o tyle ciekawa, że przyciskając lewy klawisz myszy możemy dowolnie obracać sześcianem, w którym porusza się wąż, co jednocześnie nie wpływa na przeorientowanie sterowania. A więc, po obrocie sześcianu o 180 stopni, wciskając prawy kursor poruszamy się w lewo itp.

Gra, mimo wczesnego stadium okazała się dla mnie bardzo wciągająca. Grafika jest ascetyczna i wektorowa, jednak przy tym bardzo płynna. Na chwilę obecną gra wymaga do działania Amigi z kośćmi AGA i procesora 040, w pełnej wersji z FPU. Mam nadzieję, że Phibrizzo będzie dalej pracował nad swoją grą, bo jest ona potencjalnym amigowym hitem.

Link:

<http://www.pouet.net/prod.php?which=66840>

10. Miejsce: Anarcho Ride Atari STe/Falcon (174 punkty)

Druga gra na „duże” Atari, przygotowana przez kodera Thomasa Ilga, nad grafiką pracował Alan G. Tomkins, muzykę przygotował Xfalcon. Anarcho Ride, to wyścigi samochodowe w stylu Lotus, z dodanymi elementami strzelaniny, które działają na Atari STe (zalecane 2 MB RAM) oraz na Atari Falcon. Gra zwraca uwagę dużą ilością opcji konfiguracyjnych, dostępnych etapów i tras (ukazał się już nawet datadisk z dodatkowymi etapami), a także pojazdów, które możemy prowadzić. Grafika w menu jest czytelna i kolorowa, żeby nie powiedzieć, że miejscami wręcz cukierkowa.

Sama rozgrywka nieco zawodzi, gra toczy się tylko na połowie ekranu (drugą połowę zajmuje mało estetyczna deska rozdzielcza), a oglądany przez gracza świat wygląda jakby produkcja była jeszcze niedokończona. Razi np. jednokolorowe niebo, bez żadnego cieniowania, czy chmur, mało urozmaicone pobocza i słabe sprity. Na szczęście niedociągnięcia grafiki ratuje rozgrywka, która jest całkiem przyjemna i dyna-

miczna. Ciekawym dodatkiem jest możliwość eliminowania przeciwników poprzez wystrzeliwanie do nich rakiet, przy czym każde trafienie daje nam dodatkowe punkty.

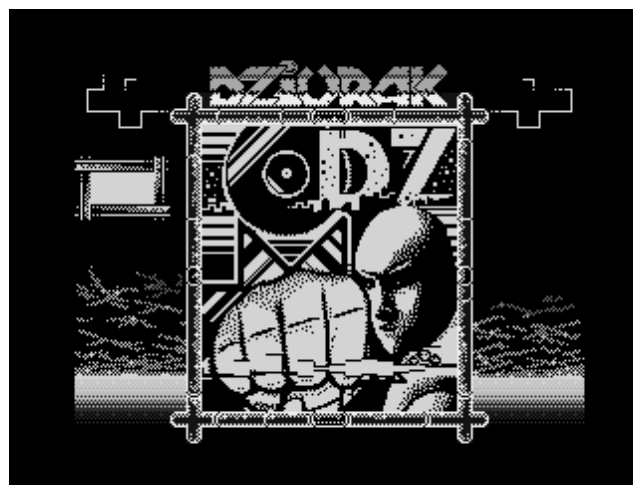
Podsumowując można stwierdzić, że Anarcho Ride jest ciekawą pozycją, jednak graficznie niewykorzystującą do końca możliwości Atari STe, o Falconie nie wspominając.

Link:

<http://www.hd-video-film.com/anarcho/info/download-anarcho-ride.php>

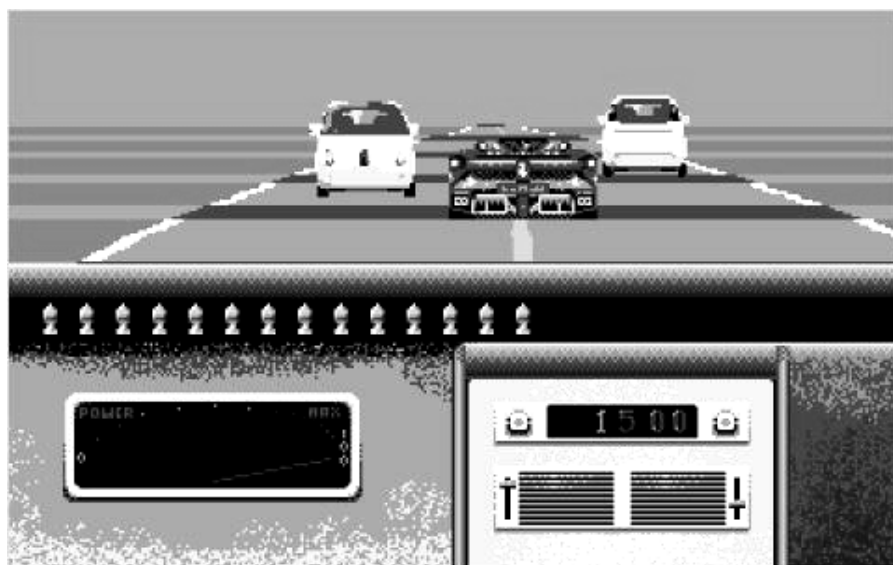
11. Miejsce: Dziurak Deluxe ZX Spectrum (172 punkty)

Grupa Specy.pl, która w 2013 zwyciężyła w pierwszej edycji konkursu z grą Krunel, tym razem zaprezentowała grę Dziurak Deluxe. Podobnie jak w 2013 roku, za grę odpowiedzialni są: Tygrys (kod ASM, poziomy), Catman (grafika, poziomy), Slayer (grafika) i V0yager (muzyka). Dziurak Deluxe jest rozbudowaną wersją produkcji, która ukazała się w 2012 roku i została wystawiona na party Wap-niak. Gra działa na ZX Spectrum wyposażonym w układ AY.



Dla ZX Spectrum też powstają nowe gry.

Celem Dziuraka jest pokonanie 40 etapów, w których musimy pokierować bohaterem, który według dostarczonego przez Tygrysa opisu, musi przeciskać się przez skalny labirynt z opadającym mu na głowę stropem. W praktyce gra polega na tym, że kierujemy w prawo i w lewo postacią u dołu ekranu, podczas gdy na górze wisi „sufit” o nierównym kształcie, zawierającym mniej lub bardziej skomplikowane uskoki. Naszym zadaniem jest takie ustawienie ludzika, żeby w momencie opadnięcia „sufitu” umiejscowiony był on w miejscu, w którym znajdzie się uskok i tym samym uniknął zgniecenia. Oczywiście wraz z pokonywaniem kolejnych etapów stają się one coraz trudniejsze, zmienia się też motyw graficzny. Muszę przyznać, że byłem nieco zaskoczony niską pozycją gry w compo, bo chłopaki w 2013 ustawili sobie poprzeczkę bardzo wysoko. Dziurak Deluxe jest całkiem wciągającą i dobrze wykonaną gierką, przy której można miło spędzić kilkanaście minut.



Ścigamy się na Atari ST.

Link:

<http://www.po-uet.net/prod.php?which=66856>

12. Miejsce:
Chase
Amiga OCS
(164 punkty)

Chase jest drugim portem gry z NES przygotowanym na konkurs przez Asmana. Podobnie jak w przypadku Alter Ego, twórcą oryginału jest Shiru. Chase to bardzo prosta i cały czas nieukończona (na chwilę obecną ma ledwie 5 etapów) gierka w stylu klasycznego Pac-Mana.

Sterujemy zieloną głową, która zbiera złote monety, przy czym musimy unikać ścigających nas „policjantów”. Gra ma przyjemną kolorową grafikę i proste efekty dźwiękowe, jednak na chwilę obecną brak jej wiele, aby móc nazwać ją grą ukończoną. Poza większą ilością leveli (w końcu to wersja preview), przydałoby się urozmaicenie rozgrywki i jakaś muzyka w tle.

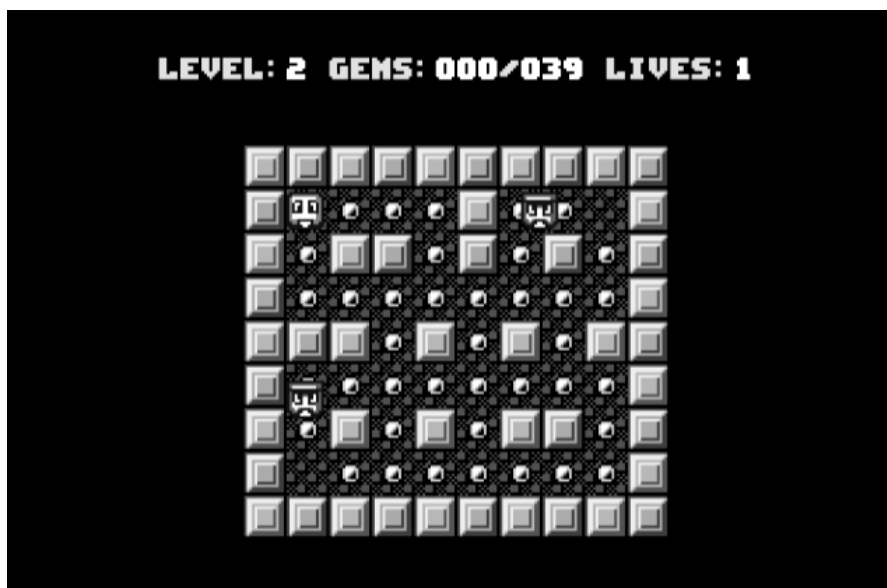
Link:

<http://www.ppa.pl/artku-ly/download/ftp/Chase-ADF.zip>

12. Miejsce:
Total Pattern
Amiga OCS
(164 punkty)

Drugi remis w RK Gamedev Compo przypadł w udziale grom amigowym. Demo logicznej układanki Total Pattern zdobyło tyle samo głosów, co Chase. Jest to kolejna gra stworzona przez forumowicza z PPA.PL, Piotra Szarejko, znanego także jako DJPiotrs. Total Pattern, to produkcja napisana w języku AMOS, w całości stworzona przez Piotra, działa na każdej Amidzie z procesorem co najmniej 020 i pamięcią Fast. Celem gry jest ułożenie klocków na planszy według zadanego wzoru, korzystając z ograniczonej liczby ruchów.

Oprawa wersji demo gry była raczej oszczędna, stąd może nie najwyższa pozycja, jednak pomysł na grę jest całkiem ciekawy i w momencie powstawania tego artykułu trwają prace nad nową wersją, z zupełnie nową oprawą graficzną i muzyką. Życzę powodzenia!



Link:

<http://amiga.net.pl/dlf/rk/totalpattern.zip>

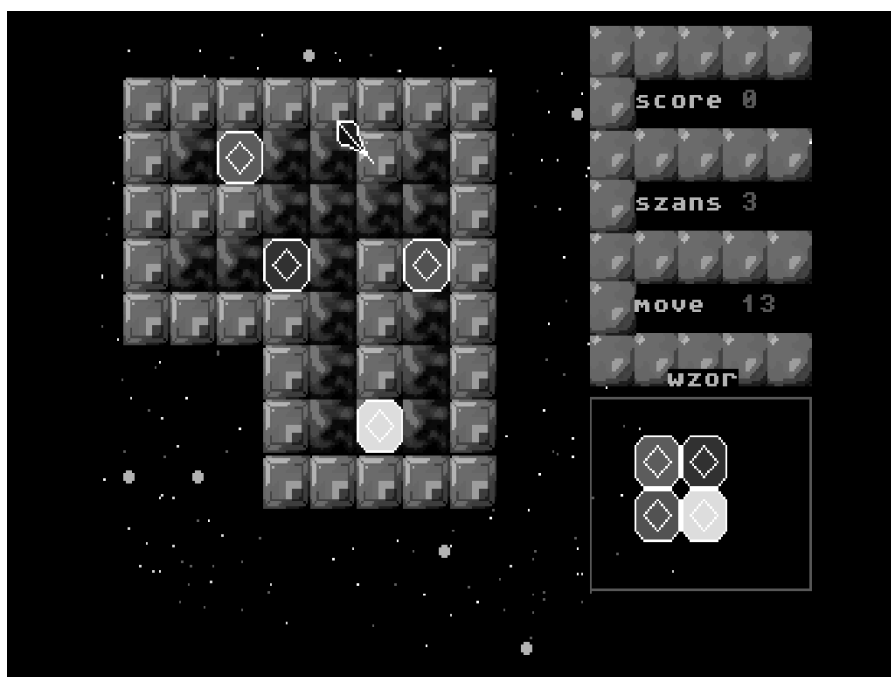
13. Miejsce:
PETSCII Poker
Commodore 64
(125 punktów)

Jak sama nazwa wskazuje jest to przygotowana na C64 symulacja gry w pokera, zaprezentowana w całości w znakowym trybie PETSCII, bardzo popularnym w ostatnim czasie na C64. Gra została przygotowana przez znaną grupę Software of Sweden. Kod gry w Basicu, wraz z wstawkami w ASM

przygotował Ron, ekran tytułowy dostarczył Shine, a za muzykę odpowiada Nvitia. Muszę przyznać, że przed compo, podczas testowania produkcji najwięcej czasu spędziłem właśnie przy PETSCII Poker.

Gra jest prosta (dla znających zasady gry w pokera), jednak bardzo wciągająca. Rozgrywkę umiła nam całkiem dobry chiptune, grafika jest, co oczywiste również bardzo prosta, ale w sumie nic więcej w takiej grze nie jest potrzebne.

Obsługa gry nie jest skomplikowana i odbywa się w całości z klawiatury. Mimo przedostatniej pozycji w compo, polecam grę każdemu commodorowemu karciarzowi.



Dwie gry zajęły ex aequo 12. miejsce.



Poker zna każdy, ale w trybie znakowym?

Link:

<http://www.pouet.net/prod.php?which=66528>

14. Miejsce: Sudoku Commodore 64 (112 punktów)

Ostatnie miejsce w głosowaniu zajęła napisana w Basicu gra Sudoku na C64. Autorem jest kolejny forumowicz z PPA.PL – RISC. Jak to w Sudoku bywa, celem jest wypełnienie diagramu 9 × 9 w taki sposób, aby w każdym wierszu, w każdej kolumnie i w każdym z dziewięciu pogrubionych kwadratów 3 × 3 znalazło się po jednej cyfrze od 1 do 9. Ja osobiście nie jestem zwolennikiem tej łamigłówki, jednak wiele osób nałogowo wypełnia kwadraciki cyferkami.

Gra napisana jest w Basicu i co oczywiste w takim wypadku działa w trybie tekstowym, a cała obsługa odbywa się z klawiatury. Mimo, że zajęła ostatnie miejsce w konkursie, to wykonana jest poprawnie i mogę ją z czystym sumieniem polecić każdemu zwolennikowi Sudoku.

Link:

<http://amiga.net.pl/dlf/rk/sudoku-c64.zip>

Podsumowanie

Druga edycja RetroKomp Gamedev Compo pokazała dobitnie, że bardzo wiele osób nadal chce tworzyć nowe gry na stare komputery. Wprowadzony przez nas wymóg, aby każda gra była napisana przez programistę, bez użycia gamedmakerów, znacznie podniósł

poprządkę. Jak się jednak okazało zgłoszeń było dużo, a ich poziom okazał się bardzo wysoki. Ostatecznie w konkursie wystartowały gry na następujące komputery i konsole: Amiga, Atari 8-bit, Atari 16-bit, Commodore 64, Commodore PET, ZX Spectrum i NES. Zgłoszeń mogło być znacznie więcej, niestety nie wszyscy autorzy zdążyli ze swoimi produkcjami.

Z tego miejsca, chciałbym podziękować wszystkim, którzy przyczynili się do sukcesu RK Gamedev Compo. W szczególności twórcom gier, którzy włożyli bardzo dużo pracy w swoje dzieła i dostarczyli je w imponującej liczbie. Bez Was nie byłoby żadnego konkursu, a praca organizatorów poszłaby na marne! Gorące podziękowa-

nia należą się także sponsorom nagród: Lotharkowi, Architect1200, Sanjyuubi i Jenotowi, dzięki którym mogliśmy uhonorować twórców czymś więcej, niż tylko uściskiem dłoni. Swoje jakże istotne „trzy grosze” dołożył nasz redaktor naczelny, Adam Zalepa, dzięki któremu gra Tanks Furry doczekała się profesjonalnego wydania.

Cichymi bohaterami natomiast są wszyscy, dzięki którym sama impreza RetroKomp/LoadError 2015 doszła do skutku i okazała się dużym sukcesem. Nie będę tutaj wymieniał wszystkich ksyw organizatorów, wiecie, kim jesteście i ile pracy włożyliście w to, żeby wszystko wypaliło.

Mateusz „Tfardy” Eckert



Sudoku prosto z Japonii na Commodore 64.



Grafika tytułowa gry z 7. miejsca.

Silly Venture 2k16

Have you played ATARI today?

Po rocznej przerwie powraca jeden z największych międzynarodowych eventów poświęconych platformom Atari. Ostatnia edycja w 2014 roku zgromadziła ok. 220 uczestników, którzy przyjechali do Gdańska dosłownie z całego świata - z Hiszpanii, Irlandii, Grecji, Szwajcarii, Finlandii, Szwecji, Danii, Francji, Wielkiej Brytanii, Niemiec, Czech, Słowacji, Holandii oraz Polski. Był to absolutny rekord jeśli chodzi o imprezę z takim targetem, ale nie tylko pod względem frekwencji - otóż do konkursów oddano ponad 120 prac w przeróżnych kategoriach: od Atari VCS, po XL/XE, ST/STe, Falcona, a tak-

że przenośną konsolę Atari Lynx czy ostatnie dziecko Jacka Tramiela - 64-bitową bestię Atari Jaguar.

Nazwa imprezy nie jest przypadkowa. "Silly Venture" (mimo swojego rozma- chu) ma bardzo luźny charakter, jest połączeniem wielu nurtów, integruje to środowisko wielotorowo - jest zarówno darmowe piwo, catering, jak też devil's show, koncerty z solidną dawką chip-tu- nowej muzyki (wykonywaną przez za- proszonych artystów z zagranicy, jak chociażby Ultrasyd czy STU) oraz wie- logodzinna prezentacja zgłoszonych do konkursów prac. Na "Silly Venture" pre- zentowane są te nowe rozwiązania sprzętowe do całej linii konsol oraz komputerów firmy Atari.



wideo) pozdrawiały już tak znane po- staci w tym środowisku jak sam twórca Atari - Nolan Bushnell, rodzina Jacka Tramiela, czy też projektant Atari ST (wcześniej C-64) - Shiraz Shivji.

Nie zabrakło osób z szeroko pojętego show-businessu - pozdrowienia od Ski- by z zespołu Big Cyc, Muńka Staszczy- ka z zespołu T-Love, Kojo, Jarka Janiszewskiego z zespołu Bielizna i Czarno-Czarni, celebrytów znanych z sieci jak pamiętny kandydat na prezy- denta miasta Białystok - Krzysztof Ko- nonowicz, Chwytak czy Pani Basia z Klatki B. Na ostatniej edycji w 2014 roku swoją obecnością zaszczytili: Sła- womir Łosowski, założyciel i klawiszow- iec zespołu KOMBI (który w swoim studio z powodzeniem używa Atari ST do dnia dzisiejszego) oraz LIROY, który za sprawą wydanej w 1995 roku i skła- danej na komputerze Atari Falcon płyty Alboom - w nakładzie przekraczającym 500 tys. egzemplarzy - osiągnął naj- większy sukces komercyjny w historii polskiego hip-hopu (obecnie jest po- słem na Sejm RP z ramienia Kukiz '15).

"Silly Venture" - nieszablony i nie- konwencjonalny event, skierowany do każdego fana produktów firmy, która ponad 20 lat była znaczącym uczestni- kiem rynku komputerowego na świecie.



Zamiast magnetofonów i stacji dysków są czytniki kart SD, twarde dyski, oraz inne współczesne nośniki danych - choć po dziś dzień są i tacy, dla których dźwięk wczytyw- nych programów z ma- gnetofonu porównywalny jest z symfonią Mozarta czy Beethovena.

O randze "Silly Venture" niech świadczy fakt, iż uczestników tej imprezy (za pośrednictwem klipów

W tym roku "Silly Venture 2k16" odbę- dzie w dniach od **11-go do 13-go listo- pada** (piątek - sobota). Jakie niespodzianki czekają na Was tym ra- zem? O tym dowiecie się już nieba- wem!

www.sillyventure.eu

**Get ready for adventure
at Silly Venture !**

Serdecznie wszystkich zapraszam.

Grey / Mystic Bytes



7

COMPACT
disc

© PROJECT R3D 2015



CDLabel.info

TANKS FURY

Project R3D przedstawia
II miejsce RetroKomp 2015 / Partner projektu INDIEGO!

Dołącz do gry i walcz o wolność!



Minimalne wymagania:

OCS, 020 procesor, 1mb pamięci chip
i 0,5 mb dowolnej pamięci

- Szybki kod, ręcznie robiona grafika pixelart
- Ciekawa muzyka i odgłosy w stylu „chiptune”
- Ponad 40 misji gry solo lub w kooperacji
- Regulowany poziom trudności
- Kilka rodzajów terenu, pustynia, las, miejska dżungla
- Pięć typów nieprzyjacielskich czołgów
- Kilka różnych zakończeń gry
- Multiplayer od 2 do 4 graczy na jednym ekranie!

ONLY
FOR!



AMIGA