



Polskie Pismo Amigowe

Numer 3/2011 (6)

cena 23 zł



Dobry Hosting w PPA.pl

Już za 100 PLN rocznie* możesz mieć:



Konto 1000 MB, 10 GB transferu miesięcznie, unikalny adres w domenie *.ppa.pl, 10 baz danych (MySQL i PostgreSQL), FTP, WWW, PHP, CGI, ColdFusion, Tomcat, CRON. Własne strony błędów, kopie bezpieczeństwa, pomoc techniczna. Do tego wiele nielimitowanych opcji: konta e-mail z ochroną antywirusową i antyspamem, z dostępem przez www, aliasy/przekierowania e-mail, parkowanie własnych domen, subdomeny. Duży wybór preinstalowanych aplikacji www (CMS-y, galerie, blogi, fora i inne), a wszystko jest zarządzane poprzez WWW za pomocą wygodnego panelu administracyjnego PLESK.

... a za dodatkowe 100 PLN Twoje konto zwiększy się do:

2000 MB pojemności
20 GB transferu miesięcznie
20 baz danych

A wszystko w domenie www.ppa.pl
- największego w Polsce portalu poświęconego Amidze i tematom z nią związanym.

W ofercie dostępne także inne warianty, również konta darmowe.

Sprawdź na www.hosting.ppa.pl

* możliwa płatność jednorazowa lub w ratach

APEL O ARTYKUŁY!

Drodzy Czytelnicy,

Czekamy na Wasze artykuły, które zasilają bazę artykułów do kolejnych numerów. Przed nadesłaniem artykułu prosimy o skonsultowanie tematu z redakcją. Artykuły prosimy nadsyłać w postaci plików tekstowych (ASCII) wraz z dołączonymi obrazkami lub zdjęciami (format PNG lub JPG). Propozycje oraz sugestie należy nadsyłać na adres podany w stopce redakcyjnej.

Konkurs na artykuł

Redakcja PPA ogłasza konkurs na napisanie artykułu do „Polskiego Pisma Amigowego”. Temat artykułu powinien pasować do profilu pisma i powinien być wstępnie uzgodniony z redakcją (np. e-mailowo na adres kontaktowy redakcja@ppa.pl). Artykuł zgłoszony do konkursu nie może być krótką notką, orientacyjne minimum to 10 000 znaków oraz ilustracje.

Redakcja zakwalifikuje zgłoszone artykuły, a następnie wybierze zwycięzcę. Zastrzegamy sobie prawo do wydrukowania każdego ze zgłoszonych artykułów w „Polskim Pismie Amigowym”. Artykuł wybrany do druku nie może być nigdzie opublikowany ani przed, ani rok po wydaniu go drukiem w PPA.

Oczywiście zwycięski artykuł ma gwarantowany druk w naszym piśmie. Na autora zwycięskiego artykułu czekają nagrody.

Oprócz tego tradycyjnie każdy autor artykułu zakwalifikowanego do druku otrzyma bezpłatny egzemplarz pisma. Termin nadsyłania artykułów upływa **30 listopada 2011 roku**.

Przy ocenie artykułów będziemy brali pod uwagę atrakcyjność tematu, styl pisania, wnikliwe podejście do opisywanych zagadnień, przygotowanie ilustracji, jak również naszą subiektywną opinię. Zapraszamy do udziału!



Polskie Pismo Amigowe

Redaktor naczelny: Sebastian Rosa.

Zespół redakcyjny: Grzegorz Kraszewski, Grzegorz Murdzek, Piotr Sadowski, Krzysztof Żegleń, Konrad Czuba.

Korekta: Aleksander Piotr Chyliński.

Skład: Sebastian Rosa

Kontakt: redakcja@ppa.pl, <http://ppa.pl>.

Skład pisma wykonywany jest w programie OpenOffice 3. Druk na urządzeniach cyfrowych firmy Ricoh.

Polskie Pismo Amigowe jest wydawane w wolnym czasie członków redakcji i autorów. Redakcja nie gwarantuje regularnego ukazywania się kolejnych numerów. Cena jaką płacisz za pismo pokrywa jedynie koszty jego wydawania, pismo nie przynosi zysków.

Poglądy wyrażane w artykułach są poglądami ich autorów i niekoniecznie odpowiadają stanowisku redakcji.

Prawa autorskie artykułów należą do ich autorów. Przedruk i publikacja w formie elektronicznej wyłącznie za zgodą redakcji. Pojawiające się w artykułach słowne i graficzne znaki towarowe firm trzecich są użyte wyłącznie w celach informacyjnych i pozostają własnością tych firm. PPA nie jest gospodarczo związane z żadną z tych firm.

© Polski Portal Amigowy 2010-2011.
Tux (maskotka Linuksa) © Larry Ewing,
Simon Budig i Anja Gerwinski.

<http://www.ppa.pl>



AmigaOS 4.1 Update 3

Z końcem sierpnia firma Hyperion Entertainment wydała trzecie uaktualnienie dla systemu AmigaOS 4.1. Przeznaczone jest dla wszystkich wspieranych platform i dostępne jest za darmo dla zarejestrowanych użytkowników systemu w tej wersji. Najważniejsze zmiany, które wprowadza uaktualnienie to:

- uaktualnione sterowniki PATA i SATA,
- ulepszona stabilność i prędkość działania DOS-a,
- poprawione błędy związane z obiektami współdzielonymi,
- obsługa klawiatur AmigaOne,
- uaktualnione komponenty GUI i Intuition,
- uaktualnione certyfikaty AmiSSL,
- uaktualniony stos USB wprowadzający obsługę USB 2.0 (EHCI),
- usprawniona obsługa Sam460ex włącznie ze sterownikiem dźwięku,
- obsługa I2C dla płyt głównych Sam440ep i Sam460ex,
- uaktualniona i ulepszona obsługa Warp3D,
- szybsze procedury grafiki 2D,
- usprawnienie systemu notyfikacji,

W numerze

	Wiadomości	3
	Odprawiamy czary, czyli jak ustrzec się przed potencjalną śmiercią Amigi .	5
	Zrób to sam – adapter CF w A600	8
	Język Lua w MorphOS-ie	9
	As wywiadu, czyli jak to się zaczęło – historia Benedykta Dziubałtowskiego .	12
	Discworld	14
	Czy Amiga może być gwiazdą imprezy?	16
	Nasza pierwsza gra – kurs programowania pod AmigaOS – część 1	17
	Sam460ex	20
	PyMUI - czyli tworzymy okna Pythonem	24
	Amiga i Manga	26
	Amiga Mods Anthology – zbiór prawie doskonały	27
	Krzyżówka, rozwiązanie konkursów	27
	Komiks	28

Od redakcji

Na początek chciałbym podziękować wszystkim, którzy mieli swój wkład w powstanie tego numeru. Naprawdę dopisała liczba artykułów, które pozwoliły bez większych trudności zbierać niezbędny materiał. Co również interesujące, coraz więcej osób odważa się pisać o nowych programach, nowym sprzęcie, nowych rozwiązaniach. To niewątpliwie cieszy i pozwala sądzić, że mimo tego, iż amigowy świat nie jest duży, na pewno nie jest martwy i jeszcze wiele przed nami do odkrycia i do opisan.

W szóstym numerze naszego magazynu każdy powinien znaleźć coś dla siebie. Mamy zarówno publicystykę, recenzje sprzętu, jak i gier, szkółki dla programistów, a także coś dla majsterkowiczów. Niewątpliwie „artykułem miesiąca” jest recenzja Sam460ex - najnowszego dziecka włoskiej firmy ACube Systems. Zainteresowanie mogą również wzbudzić aż trzy szkółki: język Lua, ciąg dalszy skryptów Pythona (tym razem PyMUI) oraz kurs programowania pod AmigaOS. Majsterkowicze, desingerzy i znawcy tuningu z całą pewnością nie ominą zarówno poradnika z cyklu „co zrobić, aby moja Amiga żyła długo i szczęśliwie”, jak i

tekstu poświęconego adapterowi CF dla A600. Gracze powinni sięgnąć na naszą „rozkładówkę”, gdzie czeka kolejna recenzja gry pod ScummVM. Tym razem jest to „Discworld”.

Dział publicystyczny w tym numerze również ma swoje obszerne miejsce. Poczytamy o zjawisku mangi na Amidze, zapoznamy się z kolekcją Amiga Mods Anthology, a także spróbujemy sprawdzić czy Amiga może być gwiazdą imprezy (niekoniecznie amigowego spotkania). Wszyscy fani wywiadów Benedykta Dziubałtowskiego z całą pewnością skierują wzrok ku stronie dwunastej, gdzie wywiadowca stał się... „wywiadowanym”. Może nie całkowicie, ale... Przekonajcie się sami.

Kolejny numer zamknięty, kolejna karta historii zapisana. Jak zwykle czekamy na Wasze artykuły, aby już w grudniu zaprezentować Wam kolejną odsłonę naszego periodyku. Zachęcamy do nadsyłania artykułów. Zapewniam, że warto. A tymczasem, zapraszam do lektury szóstego numeru „Polskiego Pisma Amigowego”.

Sebastian Rosa

- różne poprawki w kernelu wpływające na poprawę stabilności działania,
- uaktualnione MUI z nowymi elementami ułatwiającymi portowanie aplikacji korzystających z MUI 4.0.

Jak widać, ilość zmian jest spora i znacząca, zwłaszcza że wśród nowości pojawiła się długo oczekiwana obsługa USB 2.0 oraz usprawnione MUI, dzięki któremu praktycznie natych-

miast użytkownicy AmigaOS 4.1 otrzymali nową wersję przeglądarki internetowej OWB będącej portem programu z systemu MorphOS. Bazuje on na źródłach w wersji 1.9.

AmigaOS 4.1



Nowe SDK

Na początku września MorphOS-Team poinformował o wydaniu nowego SDK. Aktualizacji uległy również niektóre biblioteki systemowe takie jak *muimaster.library*, *asl.library*, *amiga-guide.library*. W skład aktualizacji wszedł również nowy, rewelacyjny edytor programisty Scribble.

<http://www.morphos-team.net/>



AmIRC powraca!

Po latach powraca AmIRC, jeden z lepszych klientów IRC na Amigę. Program jest darmowy i nie wymaga już klucza rejestracyjnego. Ekipa zajmująca się jego rozwojem poprawia błędy i wprowadza nowe funkcje. Równolegle rozwijana jest przez nich wersja dla AmigaOS 3.x i MorphOS-a. Odrębną ścieżką idzie rozwój dla AmigaOS 4.x.

<http://www.amirc.org/>
<http://www.amirc.de/>



Amiga Forever 2012

Firma Cloanto przygotowała nową edycję pakietu Amiga Forever. Wprowadzone usprawnienia obejmują narzędzie do tworzenia i edycji plików RP9, usprawniona wersja odtwarzacza z obsługą dostępu do dodatkowej zawartości plików RP9, obsługa systemu AROS i systemu prototypu Walker, nowe narzędzia konfiguracyjne. Pakiet jak zwykle dostępny jest w trzech odmianach: Value Edition, Plus Edition oraz Premium Edition. Wersja Premium zawiera wywiady z Davem Haynie oraz RJ Micaliem udzielone na wyłączność. Nowa wersja pakietu jest dostępna za darmo dla wszystkich zarejestrowanych użytkowników Amiga Forever 2011, którzy nabyli pakiet po 1 lipca 2011. Pozostali są objęci specjalnym pakietem uaktualniającym.

<http://www.amigaforever.com/>



Return to Castle Wolfenstein

Mark Olsen, znany także jako BigFoot, przygotował dla systemu MorphOS port gry „Return to Castle Wolfenstein”. Do działania gra wymaga dużych pokładów pamięci, a do instalacji niezbędna jest oryginalna płyta z grą w wersji PC. Port oraz program instalacyjny obsługują zarówno standardowe wydanie, jak i Special Edition oraz Platinum Edition. Rozgrywkę można prowadzić zarówno w pojedynkę, jak i w trybie multiplayer.

<http://bigfoot.morphos-team.net/files/morphosrtcw.lha>



AmigaOne X1000 nowe wieści

Firma A-EON Technology potwierdziła, że beta-testerzy komputera AmigaOne X1000 otrzymali już płyty główne Nemo 2.1 wraz z AmigaOS 4.1 Update 3. Dostarczony system operacyjny ma być testową bazą dla prac firmy Hyperion Entertainment nad wersją 4.2. Dodatkową informacją podaną przez A-EON jest zaadaptowanie na potrzeby AmigaOne X1000 nowego Multiboot Firmware, które posiada funkcję konfigurowalnego boot menu - do wykorzystania przez użytkowników z zainstalowanymi wieloma systemami operacyjnymi. Ta funkcjonalność sprawi, że staną się niepożądane rozwiązania typu yaboot.

Pojawienie się tej informacji zdaje się być wreszcie przerwaniem ciszy i sztucznego podsycania atmosfery przez publikację zdjęć obudowy, płyty głównej czy choćby wątpliwej jakości filmów prezentujących komputer w działaniu. Z niecierpliwością czekamy na premierę.

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



OWB 1.14

Fabien Coeurjoly przygotował kolejną wersję przeglądarki internetowej OWB. Program przeszedł zmianę nazwy i obecnie egzystuje pod nazwą Odyssey Web Browser. Wśród wprowadzonych nowości wymienić można dodanie popup menu w okienku wyszukiwania w celu wyświetlenia podpowiedzi Google, dodanie opcji polityki ciasteczek wraz z możliwością filtrowania, możliwość usuwania ciasteczek dla domeny z poziomu menadżera ciasteczek, możliwość ustawienia pełnego ekranu przy pomocy opcji w menu (dla tych, którzy nie posiadają klawisza F11), dodanie obsługi formatów FLV i OGG w ustawieniach media (domyślnie wyłączone), zaimplementowanie logiki w kodzie Curl odpowiedzialnej za umieszczanie dokumentów na Google Docs i YouTube (poprzednio należało korzystać ze spoofingu, który już nie jest obsługiwany), wprowadzenie notyfikacji dla funkcji upload. Całość została uaktualniona do źródeł silnika WebKit w wersji r91657, podobnie jak i komponenty Curl do wersji 7.21.7 oraz OpenSSL do wersji 0.9.8o (jest to cofnięcie wersji, gdyż nowsza, która była skompilowana poprzednim razem posiadała błędy). Szczegółowy opis wprowadzonych modyfikacji znaleźć można w dokumentacji.

<http://www.sand-labs.org/owb>



DvPlayer 0.76

Po trzech latach przerwy pojawiła się dla AmigaOS 4.1 nowa wersja programu do odtwarzania plików multimedialnych DvPlayer. Lista wprowadzonych zmian i poprawek jest bardzo długa, a najważniejsze z nich to:

- obsługa napisów dla płyt DVD,
- obsługa plików IFO,
- obsługa MPEG audio (także ze zmiennym bitrate (VBR)),
- obsługa plików AIFF,
- efekt „scope” w momencie odtwarzania plików dźwiękowych,
- obsługa trybu YUV240,
- przezroczystość dla skórek programu,
- implementacja nowego algorytmu pomijania klatek w plikach AVI, co skutkuje płynniejszym odtwarzaniem na wolniejszych konfiguracjach,
- wprowadzenie komunikatu o nieobsługiwaniu kodeku audio z możliwością wyszukania właściwego do jego obsługi,
- poprawienie błędów związanych ze spowolnieniami odtwarzania filmów na Sam440,
- funkcja NOSKIP działa również z plikami ASF/WMV,
- poprawienie błędów w wyświetlaniu i odświeżaniu GUI,
- poprawienie długości wyświetlanych filmów, która czasami była nieprawidłowa.

Rejestracja programu to koszt 29,45 euro. Do pobrania jest wersja demo, która posiada ograniczenie odtwarzania filmów do 7 minut oraz braku możliwości działania w trybie pełnoekranowym.

<http://dvplayer.amigarevolution.com/>



AmigaOne 500

Firma ACube Systems poinformowała na swojej stronie internetowej, że od 26 września bieżącego roku rozpoczęła sprzedaż komputera AmigaOne 500, bazującego na płycie głównej Sam460ex. Charakteryzuje się on następującą specyfikacją:

- płyta główna Sam460ex z procesorem o taktowaniu 1,15 GHz,
- 2 GB pamięci RAM,
- dysk twardy 500 GB,
- nagrywarka DVD,
- kontroler SATA 3512 na złączu PCI.

W skład zestawu wchodzi również system operacyjny AmigaOS 4.1 Update 3, myszka AmigaOne oraz naklejki na klawiaturę. Opcjonalnie można dodatkowo wyposażać komputer w kartę graficzną PCIe Radeon 4650 1 GB i/lub dysk SSD SATA2 o pojemności 64 lub 128 GB.





Odprawiamy czary

czyli krótki poradnik jak ustrzec się od potencjalnej śmierci Amigi...

Artykuł skierowany jest głównie do posiadaczy A4000D, ale pewne rzeczy są uniwersalne i warto o nich wiedzieć. Pewne czynności naprawcze wymagają wiedzy i odpowiedniego sprzętu. Autor nie ponosi odpowiedzialności za uszkodzenia powstałe przy nieumiejętnych modyfikacjach.

Stary komputer jest jak stary samochód - żeby móc cieszyć się z jego użytkowania, należy o niego zadbać i zrobić mu co jakiś czas solidny przegląd. Niby oczywiste, a jednak większość użytkowników klasycznych Amig zdaje się podchodzić do tematu z innej strony: „skoro działa, to nie ruszać”. Pytanie więc skąd biorą się na forach prośby o pomoc w stylu „bo wczoraj wszystko działało, a dzisiaj mam problem”? Niestety częstą przyczyną jest nasz grzech zaniedbania.

Każdy sprzęt elektroniczny powszechnego użytku (w tym nasze klasyki) jest, był i będzie projektowany na około 10 lat bezawaryjnej pracy. Potem powinien się zepsuć. Po prostu nie warto projektować tego typu sprzętu, który przetrwa dziesięciolecia. Przyczyn tego stanu rzeczy jest kilka:

- Pieniądże – przecież firmy produkujące sprzęt chcą zarabiać. A zarabiają najwięcej na sprzedaży nowego produktu.
- Technologia idzie do przodu i za 10 lat klient sam stwierdzi, że to, co ma jest stare i nawet jak działa prawidłowo to trzeba wymienić na nowy model z nowymi „bajerami”.
- Znane firmy nie chcą psuć sobie reputacji wśród klientów, dbają o to żeby sprzęt nie psuł się na gwarancji i żeby posłużył jeszcze trochę po gwarancji. Klient, który jest zadowolony z jednej marki, kupuje produkt tego samego producenta nie patrząc zbyt nio na cenę.
- I jeszcze raz pieniądze – produkując sprzęt na dziesięciolecia, producenci musieliby stosować znacznie lepsze komponenty, co podniosłoby końcową cenę wyrobu, czyniąc go nieatrakcyjnym dla masowego nabywcy.

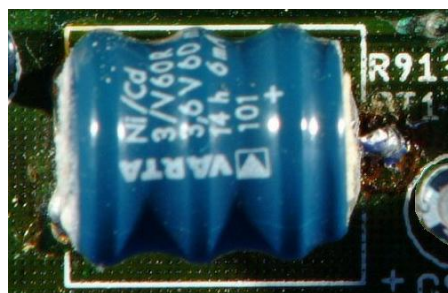
Ten sam schemat można zastosować do każdego typu produktu zmieniając tylko interwał czasowy: samochody, domy, kosmetyki... Po prostu w branży musi być ruch. Ruch to podstawa. Jeżeli jesteś fanem retro, musisz wiedzieć, że Twój sprzęt był zaprojektowany tak, aby już nie działał. To nie pomyłka – to że działa przeszło 20 lat od wyprodukowania to błąd konstruktorów. Przyznam się szczerze, że do niedawna również i ja byłem wyznawcą błędnej teorii o nieingerowaniu w działający sprzęt retro. Do czasu...

Amigę A4000D posiadam od 2002 roku i używałem jej głównie z WarpEngine 4040 + CyberVision 64, potem Picasso IV + Delfina Lite. System nie sprawiał mi żadnych niespodzianek, wszystko zawsze działało poprawnie. Jedyną rzeczą, którą zrobiłem po zakupie była wymiana akumulatora RTC (z ang. *real-time clock*) na nowy. Kłopoty zaczęły się po zamianie WarpEngine na Cyberstorm PPC oraz do-

łożeniu Mediatora 4000Di. Wszystko zaczęło się zachowywać niestabilnie. Jednego dnia wszystko działa, na drugi dzień system nie wstaje lub wstaje co jakiś czas. Próba zmuszenia Warp3D do działania na VooDoo 3 3000 kończyła się zwiem... Tak więc przyszedł czas na przegląd sprzętu. I tutaj się zaczyna cała historia, w której na moim przykładzie postaram się pokazać jak użytkownicy powinni obchodzić się ze swoim sprzętem. Co wymienić, co poprawić, gdzie upatrywać przyczyn takiego czy innego zachowania.

Akumulator RTC

Jeżeli kupujemy Amigę od nieznanego osoby, to jest to obowiązkowa rzecz do wymiany. Jeżeli komuś nie zależy na aktualnym czasie w systemie, akumulator powinien po prostu usunąć, lecz sądzę, iż jest to rzecz na tyle istotna, że warto się jednak nią zająć. Zaniedbanie tego elementu doprowadziło do śmierci już wiele Amig. Akumulator niklowo-kadmowy montowany przez Commodore w Amigach A2000, A3000D, A3000T, A4000D (nie wliczając modelu CR) oraz na rozszerzeniu A501 ma tendencję do wylewania elektrolitu po pewnym czasie. Niestety elektrolit ten jest bardzo żrący i jego wyciek na płytę zawsze oznacza jakieś uszkodzenia. Jeżeli wyciek zostanie zauważony w miarę wcześnie, to uszkodzenia są głównie kosmetyczne (przebarwienie warstwy ochronnej PCB i ścieżek miedzianych) i z reguły są niegroźne dla elektroniki. Usunięcie wylanego akumulatora, zneutralizowanie zasady kwasem ze świeżo wyciśniętej cytryny plus przemycie alkoholem izopropylowym lub spirytusem technicznym z reguły wystarcza. Gorzej wygląda sprawa z dużym wyciekiem. Najczęściej dowiadujemy się o nim, gdy po naciśnięciu przycisku „Power” nic się nie dzieje i rozkręcenie obudowy ukazuje nam okrutną praw-



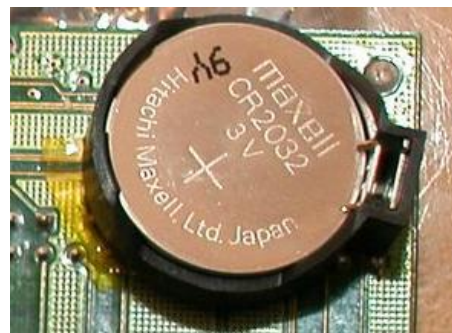
Na górze zdjęcie akumulatora RTC, na dole akumulator z wyciekiem elektrolitu



Większy wyciek

dę. Tutaj można spróbować metody soku z cytryny + przemycia alkoholem, ale z reguły w tak ciężkich przypadkach uszkodzeniu ulegają układy scalone i ścieżki miedziane na PCB i samo mycie z reguły nie wystarcza. Ponieważ procedura regeneracji nie jest tematem niniejszego artykułu, ograniczę się do stwierdzenia, że taki przypadek nadaje się tylko do naprawy.

Jeżeli już mamy usunięcie akumulatora oraz czyszczenie PCB za sobą, czas pomyśleć co dalej. Jeżeli aktualny czas w systemie jest nam zbędny, nie montujemy nic i mamy pewność, że żadnego wycieku nie będzie. Jeżeli jednak funkcja RTC jest nam niezbędna, to należy zamontować zamiennik akumulatora. Zdecydowanie najgorszym rozwiązaniem jest zamontowanie takiego samego akumulatora NiCd 3,6V – po prostu kiedyś znowu wycieknie. Lepszym rozwiązaniem jest zamontowanie baterii CR2032 w podstawce z zamontowaną szeregowo diodą.



Bateria CR2032

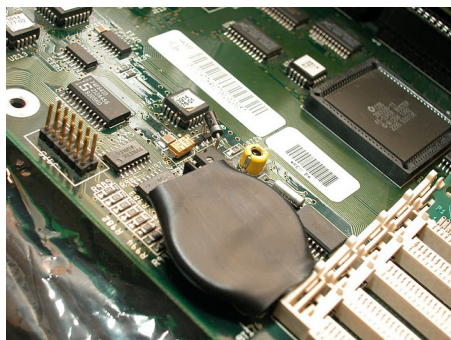
Diodę należy połączyć szeregowo z baterią. Można to zrobić na dwa sposoby:

1. Anodę diody do punktu na płycie gdzie był „minus” akumulatora, a katodę do „minusa” baterii. „Plus” baterii łączymy na płycie tam, gdzie był „plus” akumulatora.
2. Anodę diody do „plusa” baterii, a katodę do punktu na płycie gdzie był „plus” akumulatora. „Minus” baterii łączymy na płycie tam, gdzie był „minus” akumulatora.

Wadą tego rozwiązania jest konieczność zastosowania diody oraz fakt, że baterię trzeba będzie po jakimś czasie wymienić na nową.



Polecam akumulator „niecieknący” typu UL3032, zamontowany w koszulce termokurczliwej w celu uniknięcia przypadkowych zwarc. Ma on tę zaletę, że nigdy nie wycieknie, nie ma konieczności stosowania diody oraz akumulator będzie ładowany podczas pracy Amigi, czyniąc to rozwiązanie bezobsługowym.



Akumulator UL3032 zamontowany na płycie

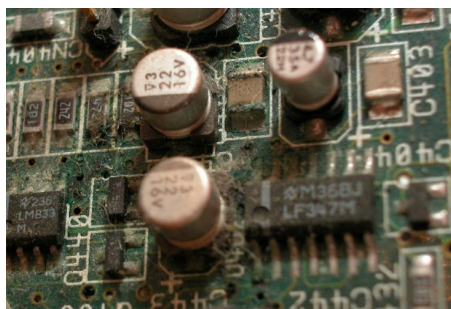
Można jeszcze zastosować akumulator (lub baterię z diodą) 3V o większej pojemności i podłączyć przewodami do płyty głównej, a sam akumulator umieścić gdzieś w obudowie z dala od płyty głównej, ale to rozwiązanie nie jest zbyt eleganckie.

Kondensatory elektrolityczne SMD

Aluminiowe kondensatory elektrolityczne SMD użyte w A600, CD32, A1200, A4000 i A4000T mają tendencję do wylewania elektrolitu. Niestety kondensatory te są tanie i kiepskiej jakości. A może to był celowy zamysł konstruktorów Commodore, żeby sprzęt padł po 10 latach i użytkownik kupił nowy sprzęt? Kto wie...

Proces wylewania elektrolitu jest jeszcze przyspieszany przez wysoką temperaturę panującą wewnątrz komputera – szczególnie w rozbudowanych konfiguracjach oraz – co najciekawsze – poprzez zły dobór typu kondensatora przez inżynierów Commodore. Na wyjściu toru audio w Amigach są dwa kondensatory elektrolityczne po jednym dla każdego kanału. Problem w tym, że najprawdopodobniej te kondensatory nie nadają się do toru audio, gdzie różnica napięć pomiędzy wyprowadzeniami nie jest duża (ok. 1V) oraz występują tam niewielkie prądy o częstotliwościach akustycznych. Te kondensatory to: C443 i C433 (22 μ F 16V) dla A4000D, C334 i C324 (22 μ F 16V) dla CD32, A1200 i A600 oraz CE320A i CE320B (22 μ F 16V) dla A4000T.

I to właśnie te kondensatory w torze audio wylewają najszybciej – szczególnie w A4000D, gdzie temperatura wewnątrz obudowy jest znaczna.



Wylane kondensatory elektrolityczne

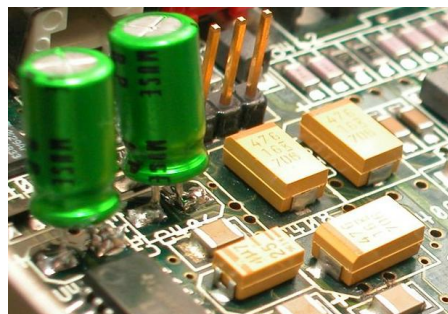
Nie wdając się w zawiłości techniczne, wylewanie kondensatorów objawia się dla użytkownika w dwójaki sposób. Po pierwsze – Amiga pracuje niestabilnie, nie zawsze się uruchomi, przypadkowe resety, brak dźwięku lub zniekształcony dźwięk. Po drugie – elektrolit jest żrący i niszczy elementy znajdujące się na płycie komputera.

Na zdjęciu widać wylane kondensatory C443 i C433 (22 μ F 16V) w torze audio A4000D. Sporo elementów jest zniszczonych. Do tego dochodzi też zniszczenie samego laminatu. Słowem – nic przyjemnego.

Radzę dla własnego spokoju Amigę rozkręcić oraz obejrzeć właśnie te „aluminiowe kubeczki” i ich okolice. Patrzymy na połączenia lutowane kondensatorów i elementów znajdujących się blisko nich. Normalnie połączenia lutowane powinny być błyszczące lub ewentualnie lekko matowe. Jeżeli zauważymy dziwne przebarwienia cyny, należy natychmiast kondensatory wymienić.

Ale co z tym robić? Czekać aż będzie gorzej i Amiga odmówi pracy? Raczej nie. Rozwiązać ten problem można tylko w jeden sposób: wymieniając wszystkie aluminiowe kondensatory SMD. I tu ważna uwaga: **kondensatory w torze audio na pozycjach podanych powyżej nie mogą być zamienione na elektrolityczne!** Po prostu te kondensatory również szybko padną. Jedyne co można zrobić, to użyć specjalnych kondensatorów bipolarnych przeznaczonych do toru audio. Są to dwa kondensatory elektrolityczne zamknięte w jednej obudowie i połączone szeregowo w taki sposób, że „plus” jednego kondensatora łączy się z „plusem” drugiego. I takimi kondensatorami zmienna polaryzacja niestraszna. Ewentualnie użyć kondensatorów ceramicznych SMD o dużych pojemnościach stosowanych w układach zasilania procesorów. Mają one dużą pojemność (5 μ F - 20 μ F), ale niskie napięcie pracy rzędu 3-6V. Dlatego jak nie znamy pełnej specyfikacji takiego kondensatora, a szczególnie jego napięcia pracy, lepiej jest zastosować kondensatory bipolarnie.

Resztę kondensatorów można zamienić na dowolne kondensatory elektrolityczne z uwzględnieniem ich napięcia i pojemności. Mogą być to nawet „zwykłe z nożkami” do montażu przewlekane. Ale najlepiej użyć kondensatorów tantalowych.



Kondensatory elektrolityczne po wymianie

Zasilacz

Ten rozdział przeznaczony jest dla użytkowników posiadających rozbudowane konfiguracje lub planujących rozbudowę. Jeżeli masz klasyczną w podstawowej lub niezbyt rozbudowanej konfiguracji – ten rozdział możesz spokojnie ominąć.

Wydawać by się mogło, że zasilacz albo działa, albo nie działa. Niestety nie jest to takie proste. Po pierwsze, zasilacz daje fabrycznie zbyt niskie napięcie. Specyfikacja A4000T (do takiej mam dostęp) mówi, że napięcia powinny być:

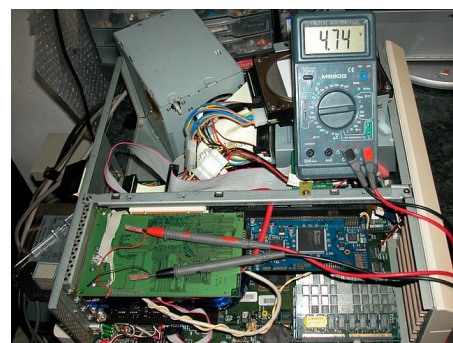
Output	+5V	+12V	-5V	-12V
Max. Load	25.0A	10.0A	0.3A	0.3A
Min. Load	5.0A	1.0A	0A	0A
Max. Voltage	5.25V	12.6V	-5.5V	13.20V
Min. Voltage	4.8V	11.52V	4.5V	-10.92V

Rozkład napięć zgodnie z „A4000T/40 Advance Engineering Release #310”

Jak widać specyfikacja napięć zasilających to:

- +5V maksimum 5,25V minimum 4,80V
- +12V maksimum 12,6V minimum 11,52V

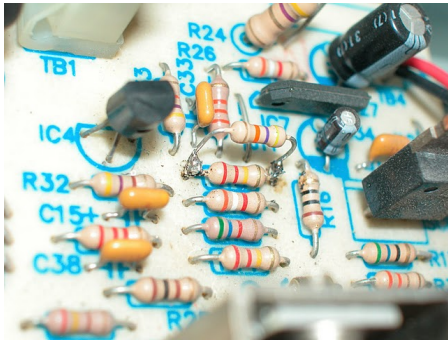
Zmierzone przeze mnie napięcie na szynie +5V przy prądzie 5A wynosiło +4,83V. Oznacza to, że dość małe obciążenie zasilacza A4000D powoduje, że napięcia wyjściowe są bardzo bliskie dolnej tolerancji. To nie jest dobre. Co więcej, na rozbudowanych konfiguracjach napięcie zasilające +5V potrafi spaść do +4,74V! Jest to napięcie średnie - przy obciążeniu procesora i karty graficznej operacjami 3D napięcie może chwilowo spaść poniżej 4,70V. A to już zdecydowanie za mało. Rezultat: Amiga robi niespodziewany reset. A my jesteśmy zaskoczeni – WTF?



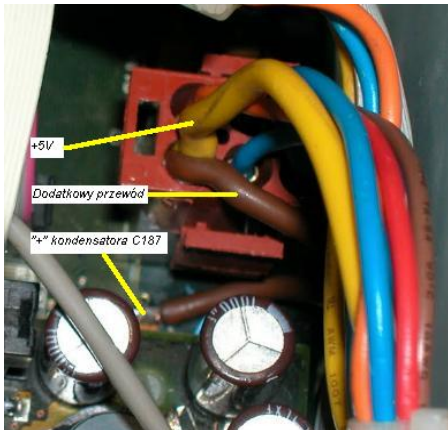
Pomiar napięcia – nie wygląda to dobrze!

Na domiar złego zasilanie +5V doprowadzone jest do płyty głównej A4000D tylko jednym przewodem! To bardzo duży błąd konstrukcyjny. W pozostałych modelach „dużych” Amig +5V jest doprowadzane do płyty trzema lub więcej przewodami. Obydwa te błędy konstruktorów należałoby naprawić, jeżeli chcemy uniknąć niespodzianek w postaci niestabilności systemu.

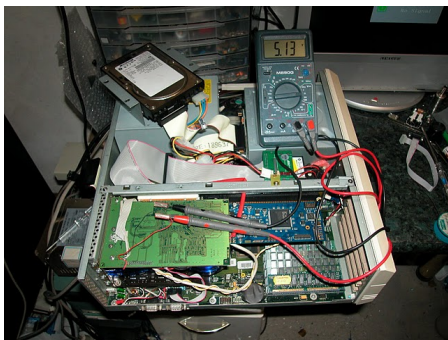
Napięcie wyjściowe zasilacza można zwiększyć lutując szeregowo rezystor 47 k Ω do rezystora 3 k Ω (lub 220 k Ω - są one połączone równolegle). Niestety na PCB nie jest zaznaczona pozycja tego rezystora. W celu lepszego rozeznania proponuję posłużyć się zdjęciem na kolejnej stronie. Problem jednego przewodu zasilającego płytę główną można rozwiązać lutując dodatkowy przewód od złotego przewodu +5V idącego z zasilacza do wyprowadzenia „plusa” kondensatora C187. Po tym zabiegu zmierzone napięcie średnie na karcie VooDoo 3 miało wartość 5,13V. Jest to nawet nieco za dużo, dlatego zamiast rezystora 47 k Ω można zastosować 68 k Ω .



Dodatkowy rezystor



Dodatkowy przewód 5V

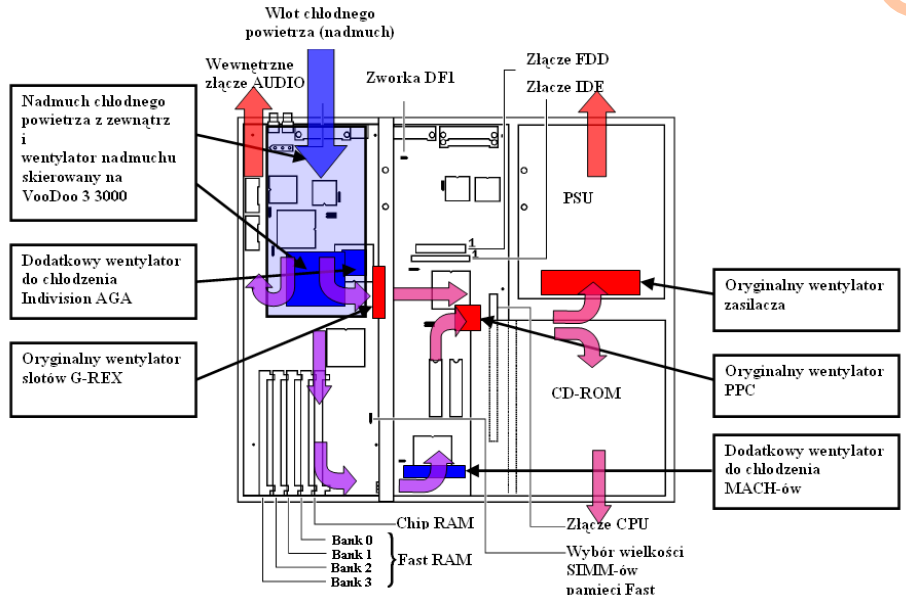


Kolejny pomiar napięcia – 5,13V to dobry wynik!

Wentylacja

Również temat interesujący dla użytkowników rozbudowanych konfiguracji. Oryginalna konstrukcja A4000D była przewidziana do karty procesorowej 68040/40 MHz, jednego twardego dysku IDE, jednego lub dwóch napędów FDD oraz kilku kart Zorro. W tej konfiguracji wentylator znajdujący się w zasilaczu w zupełności wystarcza. Problem pojawia się, gdy w tej obudowie pojawia się karta turbo Cyberstorm PPC z CyberVision PPC lub z Mediator + kartą Voodoo, a do tego jeden lub więcej dysków SCSI i CD-ROM. We wnętrzu robi się już naprawdę gorąco. Dodajmy do tego płątaninę przewodów utrudniających przepływ powietrza i mamy piekarnik. Na marginesie: ta wysoka temperatura tylko przyspiesza wylęganie kondensatorów elektrolitycznych. Co ma w tej sytuacji zrobić użytkownik? Musi jeszcze raz poprawić pracę konstruktorów. Niestety.

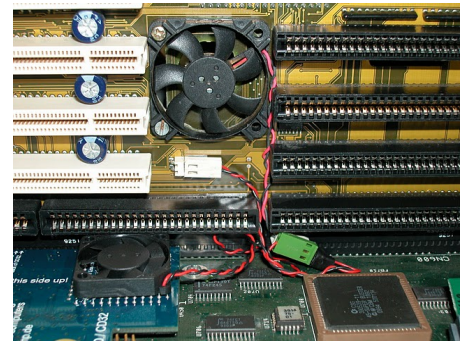
Najpierw należy przemyśleć, co tak naprawdę chcemy zrobić. Zakładanie radiatorów na wszystkie grzejące się układy (060, AGA, kontroler SCSI, układy MACH) nic tutaj nie pomo-



Przepływ powietrza A4000D

że. Po prostu ciepło cały czas pozostaje w obudowie. Jedynym skutecznym rozwiązaniem jest pozycje się ciepłego powietrza ze środka obudowy. I o tym traktuje ten rozdział.

Ponieważ wszelkie działania bez planu to chaos, rysujemy na kartce papieru PLAN. Przy jego pomocy przystępujemy do jego realizowania. Nie przejmujemy się, jeżeli coś nie będzie pasować (za duży wentylator, za mało miejsca). PLAN można modyfikować w trakcie prac. Zresztą to, co jest dobre w jednej konfiguracji, niekoniecznie sprawdzi się w innej. Nie jest istotne, jaki wentylator zostanie gdzie użyty. Ważne jest to, żeby były one dobrej jakości i nie zatarły się po roku pracy. Polecam wentylatory marki Sunon.



Chłodzenie Indi AGA i G-Rex

Słowo końcowe

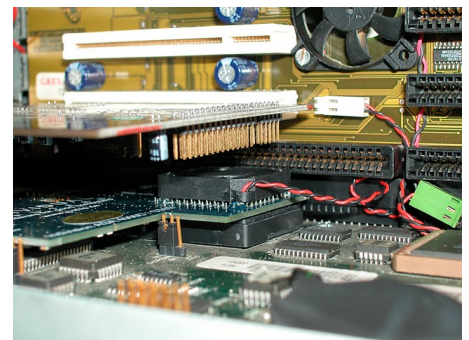
Podsumowując to, co użytkownik A4000D (i innych Amig klasycznych też) musi zrobić:

1. Zawsze sprawdzamy stan akumulatora zęgara.
2. Zawsze sprawdzamy stan kondensatorów elektrolitycznych.
3. Przy rozbudowanych konfiguracjach sprawdzamy napięcie zasilające.
4. Poprawiamy obieg powietrza, jeżeli jest to konieczne.

Wkładając nieco wysiłku na początku, będziemy mieli pewność, że zrobiliśmy wszystko, co mogliśmy dla naszej ukochanej Amigi. Nie zyskujemy pewności, że coś się nie uszkodzi samo w trakcie użytkowania - w końcu to sprzęt, który lata świetności ma już dawno to sobą. Jednak sprawdzenie tych czterech punktów pozwoli nam z czystym sumieniem stwierdzić, że zrobiliśmy wszystko, co w naszej mocy, aby sprzęt służył nam jak najlepiej. I o to tutaj chodzi.

A obok znajdują się zdjęcia przedstawiające cały mój zestaw po implementacji wyżej opisanych usprawnień.

Stanisław „stachu100” Sędłak



Chłodzenie Indi AGA i G-Rex – drugie ujęcie



Widok całości



Zrób to sam – adapter CF

w A600

W ramach przygotowania mojej Amigi 600 do montażu karty procesorowej ACA630 postanowiłem zrezygnować z dysku twardego 2.5" na korzyść karty CompactFlash. Po wszechennie dostępne na rynku adaptory IDE-CF nie spełniały jednak wszystkich moich oczekiwań, którymi są brak kolizji z ACA630, możliwość stabilnego montażu wewnątrz obudowy oraz dostęp do karty poprzez wycięcie otworu z lewej strony albo z tyłu obudowy A600. Trafił mi się nawet model adaptera współpracujący z moją A600 (ROM 37.300, płyta główna 1.3). Z pomocą przyszedł Internet, a konkretnie serwis www.amibay.com, gdzie znalazłem wątek o tytule „CF to IDE Extender, A600 and A1200” (adres do wątku w ramce). Oferowane w ogłoszeniu urządzenie jest produkowane amatorsko przez kanadyjskiego członka serwisu AmiBay o pseudonimie kipper2k i zaprojektowane zostało specjalnie z myślą o użytkownikach komputerów Commodore Amiga 600 oraz (Commodore) Amiga 1200, którzy chcą mieć możliwość zmiany karty CF bez konieczności otwierania komputera.

Adapter dostarczany jest z dwoma zworkami i wkrętem, a ze strony autora można pobrać instrukcję montażu i obsługi oraz szablony do wycięcia otworu odpowiednio dla A600 i A1200. Dostępny jest również krótki film prezentujący działanie adaptera. Urządzenie jest montowane z jednej strony do wbudowanego złącza IDE, a z drugiej strony opiera się na plastikowej ramce złącza PCMCIA, do którego jest dla zwiększenia stabilności przykręcone wspominany wcześniej wkręt. W tym miejscu istotna uwaga: kupiona przeze mnie wersja urządzenia pasuje do ramek stosowanych w A600 oraz wybranych modelach A1200, ale autor spotkał się też z takimi płytami głównymi A1200, które miały ramki złącza PCMCIA innego kształtu i w tym przypadku zaleca przyklejenie urządzenia do ramki przy pomocy kleju typu SuperGlue, aby osiągnąć zakładaną stabilność. Kolejne produkowane wersje będą odpowiednio dłuższe, aby zapewnić zgodność z większością modeli A1200.

Odnosiniki

- <http://www.amibay.com/showthread.php?t=14468> - wątek na portalu www.amibay.com,
- <http://kipper2k.com/forsale.html> - strona autora rozwiązania,
- [http://kipper2k.com/cfextender/CF Extender.html](http://kipper2k.com/cfextender/CF%20Extender.html) - film prezentujący działanie urządzenia,
- <http://classicwb.abime.net/> - ClassicWB,

Przed montażem adaptera należy nad szczeliną dla kart PCMCIA wyciąć otwór, przez który karta CF będzie wsuwana do swojego złącza. Zadanie ułatwi wspomniany wyżej szablon, który należy wydrukować i odpowiednio przyciąć przed przyłożeniem do obudowy A600/A1200. Po zaznaczeniu obrysu nowego otworu wg szablonu można przystąpić do pracy, czyli wyjęcia wszystkiego łącznie z płytą główną z obudowy Amigi i rozpoczęcia usuwania zbędnego plastiku. Zadanie jest o tyle utrudnione, że szczelina PCMCIA jest dosyć wąska, a plastik do miękkich nie należy. Autor adaptera poradził sobie, korzystając z urządzenia typu Dremel, ja natomiast posłużyłem się zestawem niewielkich pilników.

Po kilkudziesięciu minutach pracy przyszedł czas na test. Zgodnie z zaleceniem autora, przed pierwszym uruchomieniem Amigi adapter powinien pozostać pusty, a obydwa zworki muszą być otwarte. Po włączeniu zasilania zaświeci się czerwona dioda na adapterze, a Amiga wyświetli ekran powitalny ROM-u. Gdyby tak się nie stało, należy sprawdzić poprawność połączenia adaptera ze złączem IDE Amigi. Po pomyślnym teście można włożyć kartę CF do adaptera i uruchomić Amigę, np. przy pomocy dyskietki *Install*, aby przygotować kartę do pracy jak typowy dysk twardy w programie HDToolBox. W moim przypadku system nie potrafił znaleźć karty i z pomocą przyszło przestawienie dwóch zworek adaptera na zamknięte. Takie ustawienie pozwala na wyeliminowanie opóźnienia przy skanowaniu interfejsu IDE przez ROM 3.1, jeśli nie jest do niego podłączony żaden napęd. Ma to jednak pewną wadę - po podłączeniu karty dioda aktywności dysku cały czas mruga, nawet gdy nie odbywają się żadne operacje odczytu lub zapisu danych na kartę. Ignorując tę niewielką niedogodność, przygotowałem kartę CF HDToolBoxem, dokończyłem przywracanie Amigi oraz jej wnętrza do stanu wyjściowego i zainstalowałem AmigaOS 2.05.

Docelowo na kartę trafi pakiet ClassicWB, który będę mógł w wygodny sposób przenieść z Internetu na Amigę 600 przy pomocy laptopa wyposażonego w czytnik kart CF oraz z zainstalowanym emulatorem WinUAE. Wyrażone na początku oczekiwania zostały więc w pełni osiągnięte kosztem 18 CAD + 8,50 CAD za wysyłkę (~ 75 PLN).

lord_spider (PSW)



Język Lua w MorphOS-ie

Każdy użytkownik komputera po przekroczeniu pewnego stopnia zaawansowania zaczyna doceniać fakt istnienia w systemie operacyjnym języka skryptowego. W sposób szczególnie doceniają go amigowcy, albowiem ARexx - język przyjęty dla AmigaOS jako standardowy - posiada cechy wyjątkowe. Oprócz klasycznej funkcji, polegającej na automatyzacji zadań wykonywanych w linii komend, ARexx potrafi sterować aplikacjami poprzez specjalne porty komunikacyjne zwane potocznie - jakże by inaczej - portami ARexxa. W ten sposób prostym skryptem można połączyć ze sobą kilka programów tworząc z nich jeden super-program na potrzeby konkretnego zadania. Ponieważ standard się przyjął i praktycznie każdy program użytkowy na Amigę, a obecnie również i MorphOS-a jest wyposażony w port ARexxa, potęga takiego rozwiązania jest ogromna.

ARexx jednakże nie jest standardem języka skryptowego w MorphOS-ie. Przyczyna jest oczywista, jest to produkt będący elementem komercyjnego AmigaOS. Można by pomyśleć o zakupie licencji, gdyby nie fakt, że kody źródłowe zaginęły, zdaje się, w pomroce dziejów. Wiadomo też skądinąd, że ARexx był napisany w assemblerze procesora M68k. Nawet AmigaOS 4 zawiera te same pliki wykonywalne ARexxa (dla procesora M68k), co AmigaOS 2.0, w którym to język pojawił się po raz pierwszy. Mimo tego, że można uruchomić i używać ARexxa pod MorphOS-em kopiując pliki z AmigaOS, języka tego nie znajdziemy w standardowej dystrybucji systemu.

Warto też zauważyć, że napisany w 1987 roku ARexx jest implementacją języka REXX, którego historia sięga roku 1979. Przez 32 lata, jakie minęły od tego czasu, języki skryptowe i techniki programowania nie stały w miejscu. Mimo swoich niezwykłych możliwości, ARexx jako język programowania stał się językiem archaicznym. Dzięki temu, że standard portów ARexxa jest nieźle udokumentowany, istnieje możliwość rozszerzenia innych języków skryptowych o możliwość komunikowania się z aplikacjami przez porty ARexxa. Jako pierwszy w MorphOS-ie (a także w AmigaOS 4) zdobył taką możliwość popularny język skryptowy Python. Jest to jednak język dość ciężkały (interpreter Pythona zajmuje pod MorphOS-em ponad 2 MB), borykający się ze wzajemną niekompatybilnością kolejnych wersji. Nie każdemu musi się też podobać to, że składnia Pythona wymusza określone formatowanie kodu, co jest zaskakujące dla programisty znającego praktycznie każdy inny popularny język programowania.

Jeżeli nie Python, to co? Poszukując lekkiej, szybkiej, łatwej do nauczania się i elastycznej alternatywy dla ARexxa natknąłem się na język Lua. Lua ma dość egzotyczne pochodzenie - powstała bowiem w Brazylii (po portugalsku „Lua” oznacza Księżyc). Prace rozpoczęto w 1993 roku. Standard języka ustalił się wraz z wersją 4.0 w roku 2000, ostatnie większe zmiany zaszły w wersji 5.0 w roku 2003, a obecna wersja stabilna 5.1.4 pochodzi z roku 2008. Świadczy to o stabilności języka i zachęciło mnie do jego przeniesienia na MorphOS-a.



Lua jest językiem przystępnym. Tak zwana krzywa uczenia się nie jest stroma i łatwo zacząć programowanie. Z drugiej strony nie jest to na pewno język prymitywny. Po wstępnej zabawie w proste procedury i oswojeniu się z językiem, programista może sięgnąć po techniki takie jak klasy, dziedziczenie, enkapsulacja danych, przestrzenie nazw, moduły, a nawet bardziej egzotyczne techniki, np. programowanie funkcjonalne. Przy rozwiązywaniu prostych problemów można spokojnie pozostać na terenie klasycznego programowania proceduralnego. Lua jest też językiem lekkim. Biblioteka *lua.library* zawierająca interpreter liczy sobie niecałe 150 kB. Ponieważ język ten znalazł bardzo wiele zastosowań, szczególnie jako język wbudowywany w aplikacje, jest dość popularny. W sieci znajdziemy wiele stron i forów poświęconych Lua i programowaniu w tym języku, ukazało się też drukiem kilka książek na jego temat, niestety na razie bez polskich tłumaczeń.

Port

Standardowy interpreter języka Lua jest napisany w przenośnym języku C. Jego skompilowanie na MorphOS-ie jest trywialne: make, configure i gotowe. Wtedy dostajemy jednak po prostu kolejny interpreter języka skryptowego jako polecenie konsoli. Zero jakiegokolwiek integracji z systemem. MorphOS-owy port Lua ma postać biblioteki systemowej nazwanej *lua.library*. Ma to sens, bo Lua może być używana w niespotykany np. dla ARexxa sposób - mianowicie można interpreter wbudować w aplikację. Dzięki temu można w Lua pisać np. wewnętrzną logikę w grach, przetwarzać teksty, pozwolić użytkownikom na pisanie rozszerzeń do programu, tworzyć złożone pliki konfiguracyjne i tak dalej. Na innych systemach operacyjnych najczęściej linkuje się statycznie Lua z programem. Pod systemem MorphOS wystarczy otworzyć *lua.library*, dzięki temu wszystkie programy korzystają z tej samej kopii interpretera.

Kolejną ważną cechą Lua są moduły, które rozszerzają język o nowe możliwości. Moduły

można pisać w Lua, można je także kompilować np. w C. Moduły mogą być ładowane dynamicznie na żądanie, standardowy interpreter zawiera kod dynamicznego ładowania dla Windows (biblioteki DLL) i systemów unixowych (shared objects). W wersji dla systemu MorphOS zaimplementowałem dynamiczne moduły korzystające z systemowych bibliotek (*.library). Dodatkowo wszystkie moduły standardowe Lua przenieśliśmy do bibliotek zewnętrznych, co dodatkowo odchudziło interpreter. Rzadko kiedy w skrypcie potrzebne są wszystkie moduły standardowe, przez ich przeniesienie do plików zewnętrznych zużycie pamięci jeszcze się zmniejsza. Oczywiście skoro moduły Lua są systemowymi bibliotekami MorphOS-a, wszystkie skrypty ładujące dany moduł korzystają z tej samej kopii modułu w pamięci, moduł może też zostać usunięty z pamięci, gdy nie jest już potrzebny.

Lua jako zamiennik ARexxa

Ponieważ Lua bardzo łatwo rozszerza się poprzez moduły, dodanie obsługi portów ARexxa odbywa się przez załadowanie odpowiedniego modułu nazywanego *ipc.module*. Moduł ten dodaje do języka dwie nowe funkcje: *address()* i *rx()*. Pierwsza jest odpowiednikiem arexxowego polecenia ADDRESS i ustala nazwę portu, do którego będą kierowane wywołania. Analogicznie do ARexxa obsługiwany jest też wirtualny port „COMMAND”, który kieruje polecenia do DOS-a. Druga z funkcji służy do wydawania poleceń. Nie ma ona arexxowego odpowiednika. ARexx ma tę unikalną cechę, że komendy programów można pisać wprost w skrypcie, interpreter potrafi sam je ze skryptu „wyłowić”. W Lua aż tak miło nie ma, jednak nie jest to wielki problem. Oto prosty przykład, rysujący kilka kresek w programie TVPaint:

```
require("ipc")
address("rex_tvPaint")

rx("tv_SetDrawMode", "Color")
rx("tv_SetAPen", 0, 0, 0)

for i = 0, 100, 10 do
    rx("tv_Line", 100 + i, 100, 300 + i, 300)
end
```

Jak widać, w prostych zastosowaniach przejście z ARexxa na Lua nie jest trudne. Lua potrafi też oczywiście odbierać wyniki poleceń ARexxa. Obsługiwany jest zarówno wynik RC (poprzez globalną zmienną „rc” w Lua) jak i RESULT (jako wynik polecenia *rx()*), przy czym nie jest wymagane użycie polecenia OPTIONS RESULTS. Oto przykład - odczytanie wartości ARGB piksela w TVPaincie. Można je dopisać na koniec poprzedniego skryptu:

```
require("base")
require("string")

argb = rx("tv_GetPixel", 100, 101)
a, r, g, b = string.match(argb, "(%d+) (%d+) (%d+) (%d+)")
print(a, r, g, b)
```

Kod ten pokazuje też użycie modułu standardowego *string.module* do parsowania łańcuchów tekstowych. Wyniki z portów ARexxa są zawsze przekazywane jako teksty, w tym przypadku tekst zawiera 4 liczby rozdzielone spacjami. W pokazany wyżej sposób, można roz-



bić ten tekst na cztery liczby. Moduł string umożliwił przetwarzanie tekstów z wykorzystaniem wyrażeń regularnych znanych np. z Perlą, choć są tu drobne różnice.

Uważny czytelnik zauważy, że przykładowy skrypt jest nieco beztrojski, bo nie sprawdza, czy program, mający być celem wysyłanych komend, jest uruchomiony. Do sprawdzenia tego w Arexxie używaliśmy polecenia *Wait-ForPort*. Moduł *ipc.module* posiada również funkcję *waitforport()*, której używa się na przykład w taki sposób:

```
ipc.waitforport("rex_x_TVPaint", 10)
```

Po umieszczeniu tego kodu na początku skryptu, Lua będzie czekać maksymalnie 10 sekund na uruchomienie TVPainta, a jeżeli to nie nastąpi, skrypt zakończy działanie z komunikatem błędu.

Niestety Lua nie będzie prawdopodobnie w stanie zastąpić ARExxa w stu procentach, szczególnie jeżeli chodzi o starsze programy. Niektóre z nich wysyłają polecenia ARExxa bezpośrednio do portu „REXX”, nie korzystając z pośrednictwa programu *rx*. Lua mogłaby stworzyć taki port i przechwytywać polecenia zakładając, że są one napisane w Lua. Takie rozwiązanie być może miałoby szansę działać np. z programem AmIRC (polecenia definiowane w zakładce „Zdarzenia”). Niektóre inne aplikacje mogą jednak wysyłać polecenia, które nie są edytowalne przez użytkownika. Wtedy nie dość, że nie zostałyby rozpoznane przez interpreter Lua, to niemożliwe byłoby jednocześnie używanie Lua i ARExxa. W systemie nie mogą istnieć dwa porty o tej samej nazwie. Pewnym, nieco hakerskim rozwiązaniem, byłaby bezpośrednia modyfikacja (edytorem szesnastkowym) programu *RexxMast* tak, aby tworzył port o innej nazwie, np. „REX2”. Wtedy Lua przechwytywałby polecenia do swojego portu „REXX”, a jeżeli byłyby niezrozumiałe, przekazywałby je dalej do wykonania ARExxowi. Być może kiedyś zaimplementuje takie rozwiązanie.

Użycie Lua wewnątrz programu

Lua używany jako zamiennik ARExxa, komunikuje się z programem niejako z zewnątrz. Nic nie stoi jednak na przeszkodzie, żeby pisać program, np. w C, jego część napisać w Lua, tworząc wewnątrz programu instancję interpretera. Jakże mogą być powody takiego działania? Są problemy programistyczne, które w Lua może być po prostu wygodniej rozwiązać. Przykładem mogą być bardziej złożone operacje na dynamicznych tekstach. Fragment w Lua można też ładować z pliku, co pozwoli użytkownikowi na zmianę działania programu. Takie podejście wykorzystuje (niestety na razie nie w pełni ukończony) program do IRC – *Sermonatrix*. Reakcję programu zarówno na komendy własne, jak i pojawiające się na kanale IRC teksty użytkownik może oskryptować w Lua.

Wymiana danych między kodem w C (czy innym języku) a skryptem Lua jest bardzo prosta i wykonywana za pomocą funkcji z biblioteki *lua.library*. Parametry dla funkcji w Lua są umieszczane na specjalnym stosie (nie należy go mylić ze sprzętowym stosem procesora) tworzonego przez Lua. Wyniki funkcji (których w Lua może być wiele) są, po jej wykonaniu, do wzięcia z tegoż stosu. Dla przykładu rozwiążmy następujący problem. Mamy łańcuch

tekstowy zawierający działanie matematyczne, na przykład:

„62 - 71 * 12 + 102 / 3”.

Działanie to jest za każdym razem inne, wpisywane przez użytkownika. Trzeba wyznaczyć jego liczbowy wynik. Napisanie tego w C jest oczywiście możliwe, niemniej z pewnością zajmie trochę czasu. W Lua zadanie można rozwiązać na przykład tak:

```
require('string')
require('base')

function muldiv(op1, oper, op2)
    if (oper == '*') then x = op1 * op2
    else x = op1 / op2
    end
    return x
end

function addsub(op1, oper, op2)
    if (oper == '+') then x = op1 + op2
    else x = op1 - op2
    end
    return x
end

function kalkulator(wyr)
    repeat
        wyr2 = string.gsub(wyr, '(%-?%d+)%s*([%/])%s*(%-?%d+)', muldiv, 1)
        brak_zmiany = (wyr == wyr2)
        wyr = wyr2
    until brak_zmiany
    repeat
        wyr2 = string.gsub(wyr, '(%-?%d+)%s*([%+-])%s*(%-?%d+)', addsub, 1)
        brak_zmiany = (wyr2 == wyr)
        wyr = wyr2
    until brak_zmiany
    return tonumber(wyr)
end
```

Funkcja *kalkulator()* działa w ten sposób, że wyszukuje kolejne działania od lewej do prawej i każde znalezione działanie elementarne wraz z jego argumentami zastępuje jego wynikiem. Aby zachować kolejność działań, w pierwszej pętli przeglądane są mnożenia i dzielenia. Po ich eliminacji – dodawania i odejmowania. Ostatecznie po eliminacji wszystkich działań pozostaje jedna liczba, czyli końcowy wynik. A oto jak wygląda załadowanie, przekazanie parametru, wykonanie i odebranie wyniku z poziomu programu w języku C:

```
if (LuaBase = OpenLibrary("lua.library", 50))
{
    LuaState *L;
    struct LuaMemoryData lmd;

    if (L = LuaNewState(NULL, NULL))
    {
        char *wyrazenie = "8 - 7 * 2 + 10 / 4";
        LONG wynik;

        lmd.Buffer = Skrypt;
        lmd.Length = sizeof(Skrypt) - 1;

        if (LuaLoad(L, LUA_READER_MEMORY, &lmd, "kalkulator") == 0)
        {
            LuaPCall(L, 0, 0, 0);
            LuaGetGlobal(L, "kalkulator");
            LuaPushString(L, wyrazenie);
            LuaPCall(L, 1, 1, 0);
            wynik = LuaToInteger(L, -1);
            Printf("%s = %ld\n", wyrazenie, wynik);
        }
        else PutStr(LuaToString(L, -1));

        LuaClose(L);
    }
    CloseLibrary(LuaBase);
}
```

Po klasycznym otwarciu biblioteki, funkcja *LuaNewState()* tworzy interpreter. Następnie funkcja *LuaLoad()* ładuje skrypt ze zwykłej tablicy typu char, w której umieszczony jest skrypt Lua pokazany wyżej. Pierwsze wywołanie *LuaPCall()* wykonuje nam cały skrypt. Ponieważ jednak zawiera on wyłącznie definicje funkcji, to wykonanie spowoduje tylko załadowanie modułów zewnętrznych wyszczególnionych przez *require()* oraz zapamiętanie funkcji. Kolejnym krokiem jest wrzucenie na stos Lua funkcji *kalkulator()*, a następnie jej parametru, czyli wyrażenia do obliczenia. Drugie wywołanie *LuaPCall()* wykonuje naszą robotę. Dwie

jedynki jako argumenty oznaczają, że przesyłamy jeden argument do Lua i odbieramy jeden wynik.

Oczywiście - jeżeli obliczenie wyrażenia będziemy wykonywać wielokrotnie, powtarzać trzeba tylko cztery linie kodu, począwszy od *LuaGetGlobal()*. Co więcej, jeżeli zachodzi taka potrzeba, można w jednym skrypcie umieścić wiele funkcji i po jego interpretacji wywoływać te funkcje wedle uznania. Gdybyś-

my z jakichś względów woleli ładować skrypt z pliku (np. aby umożliwić użytkownikowi jego modyfikację), w kodzie zmieni się tylko tyle, że w *LuaLoad()* pojawi się stała *LUA_READER_FILE*, a zamiast wskaźnika na strukturę *LuaMemoryData* opisującej bufor w pamięci, damy po prostu ścieżkę do skryptu. Nic

nie stoi na przeszkodzie, aby w takim skrypcie wewnętrznym aplikacji wysłać komendy ARExxa do innych programów. Możliwości, jakie się otwierają, są nieograniczone

Warto się przyjrzeć ile „kosztuje” takie użycie skryptu Lua, w sensie zapotrzebowania na pamięć i szybkości wykonania. Wstępną ceną uruchomienia skryptu jest załadowanie do pamięci biblioteki *lua.library* (150 kB), *base.module* (20 kB) i *string.module* (22 kB) - razem 192 kB pamięci. Są to jednakże biblioteki

współdzielone, zatem jeżeli zostały wcześniej otwarte przez inny program, żadna dodatkowa pamięć nie zostanie zajęta. Zużycie pamięci przez interpreter wykonujący skrypt można precyzyjnie zmierzyć programem *Lua Explorer*, który mierzy na bieżąco w czasie wykonywania skryptu aktualne i szczytowe zużycie pamięci. Dla skryptu kalkulatora szczytowe zużycie wynosi 14 kB, uwzględnia to pamięć niezbędną na załadowanie skryptu i przetłumaczenie go na kod „maszyny wirtual-

nej” Lua.

Jeżeli chodzi o szybkość wykonania, to oczywiście kod w Lua będzie znacznie wolniejszy niż procedura w C. Jeżeli jednak przetwarzane wyrażenie jest wprowadzane przez użytkownika, to kwestia czy będzie ono obliczane przez jedną milisekundę, czy 20 milisekund ma znaczenie wtórne. Sprawa przedstawiałaby się inaczej, gdyby nasz program musiał wczytywać z pliku i przetwarzać setki tysięcy takich wyrażen, wtedy z pewnością dodatkowa praca włożona w napisanie parsera w języku C, przyniosłaby efekt.



Lua Explorator

W przypadku używania fragmentów w Lua w programie kompilowanym zawsze idziemy na pewien kompromis – ułatwiamy sobie i przyspieszamy pisanie programu, kosztem szybkości jego wykonania i wymagań pamięciowych. Wymagania pamięciowe nie są jednak, jak pokazałem powyżej, ogromne, natomiast szybkość wykonania nie zawsze jest sprawą krytyczną. W wielu przypadkach jest to cena, jaką warto zapłacić.

Plany na przyszłość

Port Lua dla MorphOS-a znajduje się wciąż w wersji 0. Oznacza to, że jakiś rozwój jest na pewno planowany. Przede wszystkim trzeba przeportować brakujące jeszcze moduły standardowe Lua – *math*, *os*, *table*, *coroutine* i *debug*. Pracy wymaga również Lua Explorator. Nieco bardziej ambitnym zamierzeniem jest umożliwienie pisania w Lua kompletnych programów używających MUI. Chodzi przy tym nie tylko o proste programiki bazujące jedynie na notyfikacjach. Bardzo ważne jest umożliwienie pisania w Lua własnych klas MUI, bo tylko w ten sposób można w pełni wykorzystać potencjał MUI i jego możliwości. Przydatne będzie z pewnością udostępnienie poprzez moduły innych systemowych bibliotek, chociażby *locale.library*.

Wierzę w to, że za jakiś czas Lua będzie językiem, w którym da się zrobić wszystko to, co w C, oczywiście z wyjątkiem rzeczy „blisko systemu” takich jak biblioteki współdzielone czy sterowniki urządzeń. Różnica będzie też w wydajności, wszak Lua jest językiem interpretowanym. Istnieje jednak prężnie się rozwijający projekt LuaJIT – interpreter Lua używający technik dynamicznej kompilacji. Jest on tak

napisany, że można łatwo zastąpić nim standardowy interpreter. Dynamiczna kompilacja – w dużym uproszczeniu – polega na tym, że fragmenty programu wykonywane więcej niż raz, są tłumaczone na kod maszynowy procesora i zapamiętywane. Przy kolejnym wykonaniu fragmentu nie następuje już interpretacja programu Lua, zamiast tego procesor wykonuje swój natywny kod z pełną prędkością. Świetnie dla MorphOS-a się składa, że LuaJIT obsługuje również procesory PowerPC. W przyszłości z pewnością przymierzę się do zastosowania LuaJIT w morphosowym porcie języka.

Grzegorz Kraszewski

Odnosińki

- <http://www.lua.org> - Strona domowa języka Lua,
- <http://www.lua.org/manual/5.1/> - Podstawowa dokumentacja i specyfikacja języka,
- <http://lua-users.org/wiki/> - Baza wiedzy o Lua rozwijana na zasadzie wiki,
- <http://www.lua.org/pil/> - Pierwsze wydanie książki „Programming in Lua” w wersji elektronicznej
- <http://lua.org.pl> - Strona o Lua po polsku, znajduje się tu również niedokończone tłumaczenie specyfikacji języka,
- <http://morphos-files.net/download/Lua> - najnowsza wersja Lua dla MorphOS-a.



As wywiadu

czyli jak to się zaczęło – historia Benedykta Dziubałtowskiego

Będąc na wielu amigowych zlotach, często bywam pytany o to - jak się robi wywiady oraz, jak to się zaczęło. Proszony jestem o podanie rodowodu swojego imienia i nazwiska. Pomyślałem, że napiszę o tym artykuł - aby szersze grono ludzi, którzy nie uczęszczają na amigowe spotkania, mogło o tym przeczytać.

Zacznę może od mojego imienia i nazwiska. Benedykt to moje imię z bierzmowania. Długo szukałem jakiegoś ciekawego imienia na tę okoliczność. Zdecydowana większość kumpli wybierała Piotr, Paweł, Mateusz. Ja chciałem mieć oryginalne. Padło na Cypriana. Niestety ktoś inny z ówczesnej grupy przyszłych „bierzmowańców” zdecydował się na takie imię. Dwóch Cyprianów jednocześnie to dla mnie było za dużo. Tak więc zacząłem wertować żywoty świętych i moją uwagę przykuł właśnie św. Benedykt. Takie też imię wybrałem, a było to 3 maja 2001 roku. Nazwisko to zupełnie inna bajka. Opiszę je jednak dosyć enigmatycznie z pewnych względów, do których nie chciałbym wracać pamięcią. Mimo upływu lat to jednak nadal boli.

Korzenie Dziubałtowskiego sięgają wydarzeń z rejonów opolsko-wrocławskich, a konkretnie dnia 6 czerwca 2005 roku. Właściwie to po dziś dzień żałuję, że wybrałem to nazwisko. Najbardziej chyba żałowałem 2 sierpnia 2008 roku. Ale były też czasy, że bardzo je lubiłem. Swego czasu nawet byłem w urzędzie z wnioskiem o zmianę nazwiska na Dziubałtowski, ale ostatecznie stwierdziłem, że to jest wyjątkowo głupi pomysł. Tak czy siak, nazywam się Benedykt Dziubałtowski. Imię i nazwisko jest moje. W pewnym okresie życia musiałem się z tym pogodzić i tak już raczej pozostanie. Ktoś mi kiedyś zaproponował skrócenie Dziubałtowskiego o parę liter. Wychodziłoby Dziuba, ale to jeszcze głupszy pomysł niż Dziubałtow-



ski. Może kiedyś zmienię nazwisko na Klerik? Po czesku to kancelista, czyli pisarz. Chyba nawet do mnie pasuje. Jako ciekawostkę powiem, że Dziubałtowski został kiedyś wyceniony na 10 tysięcy PLN. Zaproponowano mi taką sumę w zamian za zaprzestanie używania tego nazwiska. Niestety ja głupi nie skorzystałem. Najgorsze jest to, że propozycja nie padła ponownie. Dziś bym się nie zastanawiał.

Co do wywiadów. Historia jest dosyć ciekawa. W okolicach roku 2006 wertowałem sobie internet w poszukiwaniu artykułów o Amidze. Wpadłem przypadkiem na PPA. Zacząłem czytać forum i niejaki MDW pisał o tym, jak programował zapis stanu gry do swojej produkcji „Miało Śmierci”. Szczeka mi opadła, a oczy wyskoczyły z oczodołów. Zawsze marzyłem, aby móc porozmawiać z Twórcą gry komputerowej (przed duże T i małe g). Zacząłem szukać jakichkolwiek namarów na wtedy jeszcze bliżej mi nieznanego pana o ksywce MDW. Po wysłaniu kilku maili dostąpiłem zaszczytu rozmowy z Mariuszem Włodarczykiem. Możecie się śmiać, czytając te słowa, ale ja wtedy byłem maksymalnie „podjarany” możliwością rozmowy z „bóstwem”. Pamiętam jak dziś naszą pierwszą rozmowę na GG. Zaczęła się około godziny 18:00, a skończyła gdzieś w godzinach rannych. Mi jednak było mało. Wypytałem Mariusza chyba o wszelkie możliwe rzeczy dotyczące jego gier. Po kilku dłuższych rozmowach zaproponowałem wywiad oraz publikację jego gry „Rycerze Mroku” na PPA. Sprawa wbrew pozorom nie była łatwa. Należało uzyskać zgodę na publikację gry od wydawcy, firmy L.K. Avalon, a ten zgodzić się nie chciał. W takich sytuacjach musi przyjść ktoś, kto nie wie o tym, że tego zrobić się nie da i to musi zrobić. Tak więc, że ja o tym nie wiedziałem i z

uporem osła zacząłem małym, kieszonkowym wiertelkiem wiercić dziurę w kolanie szefa Avalonu. W końcu otrzymałem upragnioną zgodę, zrobiłem wywiad z MDW i napisałem recenzję „Rycerzy Mroku”. Cały materiał powędrował na PPA. MDW kręcił z politowaniem głową, że komuś się chce grzebać w tych wykopaliskach. Mnie się chciało i mi się to podobało.

Jednak jak dla mnie to było nadal mało. Postanowiłem zacząć szukać ludzi, którzy kiedyś tworzyli gry na Amigę oraz starać się o zgodę na ich publikację. Wiele osób pytało mnie o metody, jakimi się posługuję. Właściwie to nie było nic skomplikowanego. Na początku wpisywałem czyjeś nazwisko w wyszukiwarkę GG albo w Google. Z czasem pojawiła się „Nasza Klasa” i właśnie tam udało mi się znaleźć najwięcej ludzi. Chyba najwięcej trudu zajęło znalezienie ks. Jana Pikula. W pierwszej fazie poszukiwań znalazłem informację o tym, że ks. Jan Pikul nie żyje. Okazało się jednak, że nie o tego Pikula chodzi. Obdzwońnięm połowę polskich parafii (jeśli mnie pamięć nie myli, to coś około 60) zanim wpadłem na jego trop. Jednak ksiądz Jan Pikul odmówił wywiadu i stwierdził, że na to potrzebna jest zgoda biskupa. Jednak i to się załatwiło.

Jednym z najbardziej nieoczekiwanych wywiadów był wywiad z Markiem Pampuchem. Pamiętam temat na forum PPA „co się stało z Markiem Pampuchem”. Ludzi pytali, kombinowali. No w sumie ja też. Tylko, że ja wziąłem do ręki książkę telefoniczną, zadzwoniłem i zapytałem.

Gdy ludzie pytają mnie o wywiad, który uważam za najlepszy, to sądzę, że będzie to wywiad z Tomkiem Tomaszkiem - członkiem gru-



25/10/2008



py World Software, twórców „Franko” i „Doma-na”. Wywiad jest bardzo długi, a jego przeprowadzenie trwało w przybliżeniu półtora roku. Jednak miałem okazję wypytywać programistę dosłownie o wszystko, co tylko było możliwe. Dzięki mojemu oślemu uporowi oraz kieszonkowej wiertarce, której użyłem po raz kolejny wierząc dziurę w kolanach pozostałych członków grupy World Software, udało się także opublikować wersję nieocenzurowaną „Franka” oraz całkiem sporo materiałów z nigdy niewydanej drugiej części gry.

Robienie wywiadów wiąże się z wieloma profitemi. Pewnego razu dostałem od kogoś piwo Karmi, a na jednym z Amizaduszek nawet baton Bounty. Ale takim niematerialnym profitem było chyba członkostwo w redakcji PPA. Rozpierała mnie duma, gdy dzwoniłem do kogoś i mówiłem „Dzień dobry, z tej strony Benedykt Dziubałtowski. Jestem jednym z redaktorów Polskiego Portalu Amigowego, czy zgodzi się Pan na wywiad?” Warto nadmienić, że zrobiono nawet o mnie grę na Amigę. Za 20 lat będzie czym się dzieciom pochwalić. Oprócz tego bardzo miła atmosfera jest na zlotach amigowych. Dzięki mojej pracy stałem się dosyć rozpoznawalną osobą w amiświatku. Efektem tego była nawet piosenka zaśpiewana

dla mnie na jednej imprezie AmiWawa. A jak ludzie reagowali na takie powitanie? Bardzo różnie, ale w zdecydowanej większości pozytywnie. Jedną z moich ofiar wywiadowczych nawet zaczęła krzyczeć głośno do słuchawki, a po chwili dodała „O Ku**a, to ktoś jeszcze o tym pamięta?” Takich i innych miłych zdarzeń było dużo więcej. Zdarzały się też oczywiście mniej miłe momenty, typu wyzwiska i tym podobne, ale nie będę się na ten temat rozwodził. Nie ma sensu, lepiej pamiętać dobre chwile, takie jak choćby to, gdy wpadłem jak parowóz na ślub wspomnianego już wcześniej MDW. Ślub odbył się we Wrocławiu. W rękę trzymałem dyskietkę z grą, która MDW popelił - „Pechowy prezent” oraz przygotowałem dla niego życzenia, w które wplecione były tytuły jego gier. Reakcja MDW była, powiedzmy, powściągliwa. Ale sądzę, że ostatecznie mój dziwaczny pomysł mu się spodobał.

Na sam koniec chyba najdziwniejsza rzecz. Klika osób było pewnych, że jestem po dziennikarstwie albo w trakcie lub po prostu mam mgr przed nazwiskiem. A tu kłops. Z wykształcenia jestem elektromechanikiem ze specjalizacją „przewijanie silników elektrycznych”. Przed nazwiskiem zamiast mgr mam czeladnik. Tak, to prawda. Jestem po zawodowce.

Cztery razy próbowałem zdobyć średnie wykształcenie, bo marzyły mi się studia. Niestety - cztery razy szkołę oblałem (ostatni raz w grudniu 2010). Chyba nie jest mi to pisane. Braki w edukacji jednak nie przeszkadzają ani w prowadzeniu mojej firmy, ani w amigowaniu, ani w robieniu wywiadów.

Na sam koniec opiszę najgłupszą sytuację w mojej amigowej karierze. Zdarzyło się to 1 czerwca 2011 roku. Pojechałem z moją Amigą do Mastera, aby zreanimować s-video. Reanimacja się powiodła, niestety ustawiony overscan nie trzymał synchronizacji na telewizorze Mastera. Master więc odpałił moją Amigę bez sekwencji startowej i zanim załadował *setpatcha* i *loadwb* chciał przypisać ENV do RAM. Wpisał więc w CLI „*mout env: ram:*”, a ja zwróciłem mu uwagę, że w składni brakuje literki „n”. Master poprawił, ale Amiga nadal nie przypisywała prawidłowo ENV tylko wyrzucała jakieś błędy w katalogu DOSDrivers. Kombinowaliśmy nad tym parę dobrych minut. Master kasował dwukropki, a ja kombinowałem co jest nie tak. W pewnym momencie mnie oświeciło i uświadomiłem sobie kretynizm sytuacji. Aby pomylić *mount* z *assign*, to trzeba mieć nieźle wypite. Najgorsze jest to, że obydwaj byliśmy trzeźwi.

Tym optymistycznym akcentem chyba skończę mój wywód. Wypadałoby chyba kogoś pozdrowić. No więc pozdrawiam moją Marcelinę, która co prawda Amigę nie rozumie i pyta mnie, czym to się różni od komputera, ale na szczęście akceptuje.

Benedykt Dziubałtowski



Benedykt Dziubałtowski z siostrą – rok 1989





Discworld

Oto wielki A'Tuin - gwiazdny żółw, sunący majestatycznie przez otchłań kosmosu. Na swej skorupie niesie cztery słonie, dźwigające niewielki, dyskowaty świat, otoczony bezustannie przelewającym się w nicość oceanem. Przesuwają się przed naszymi zdumionymi oczyma, a plumkająca w tle muzyczka MIDI... Zaraz, zaraz! Jaka muzyczka?! Ach, to dlatego, że nie czytamy jednej z książek z cyklu „Świat Dysku”, tylko oglądamy intro do gry przygodowej „Discworld”.

W czym rzecz?

Książki ze „Świata Dysku”, których autorem jest Terry Pratchett, cieszą się niesłabnącą popularnością wśród miłośników „lekkiej fantastyki”. Napisane są w niepowtarzalnym stylu i z olbrzymim poczuciem humoru. Stworzenie na ich podstawie gry komputerowej było tylko kwestią czasu. A uruchomienie jej na naszych Ami-maszynach... cóż, było tylko kwestią nieco dłuższego czasu. Nie ukrywam, że czytałem większość książek ze wspomnianego cyklu i sprawiło mi to niemalą frajdę. Gdy więc zobaczyłem na liście obsługiwanych przez ScummVM gier znajomy tytuł, nie posiadałem się z radości. Zwłaszcza że gdzieś w zakamarkach mojej pamięci tkwiło wspomnienie zachęcająco wyglądających, zgrabnie narysowanych, dużych postaci, jakie w dzieciństwie widziałem na ilustracjach w którymś z „Secret Servisów” (okazały się potem ilustracjami z drugiej części, ale o tym później). Drżącymi z przejęcia rękami uruchomiłem ScummVM, a następnie wybrałem katalog z zainstalowaną wersją z PC. Oczom moim ukazało się znajome logo Psygnosis. Po nim przyszła kolej na kilka napisów, aby w następstwie na ekranie monitora zagościło wspomniane wprowadzenie. Po jego zakończeniu było już jasne, że gra bazuje na jednej z książek cyklu, a konkretnie na „Straż, straż!” (którą akurat przyszło mi czytać jako pierwszą). A jak to wszystko wyszło?



„Szanowna wycieczko, oto typowa łazienka w stylu Ankh-Morporiańskim. Proszę zwrócić uwagę na piękne odwzorowanie efektu rampy”

„Nauka Świata Dysku”

Grę obsługujemy za pomocą myszki. Lewym przyciskiem przemieszczamy Rincewinda - głównego bohatera - po planszy oraz otwieramy inwentarz. Lewym przyciskiem wybieramy także dostępne w rozmowie opcje dialogowe (podobnie jak w „Sam&Max Hit the Road” symbolizowane za pomocą szeregu ikon). Wykonując dwuklik zbieramy przedmioty i używamy ich na innych, w tym także łącząc przedmioty w inwentarzu. Z kolei prawy przycisk wykorzystujemy do oglądania elementów scenarii i obiektów. Słowo komentarza odnośnie naszego inwentarza. Są dwa: niewielki, na który składają się nabyte umiejętności i rzeczy, które nosimy przy sobie oraz drugi, większy - dostępny, gdy już uda nam się dobrać swój ruchomy bagaż.

„Ruchome Obrazki”

Przykro mi o tym mówić, ale grafika w grze kompletnie do mnie nie trafia. Na pewno (i wielka szkoda) nie jest to charakterystyczny styl Josha Kirby'ego, w którym utrzymana jest większość okładek książek z cyklu. Nasz bohater, Rincewind, mag-nieudacznik wygląda żałośnie, porusza się dziwnie sztywnie i - chociaż wyposażony w pokaźny garnitur ruchów - wykonuje je nadzwyczaj mało płynnie. Dostępne scenerie wahają się od nawet ładnych, do absolutnie paskudnych. Wykonano je skanując odręcznie wykonane rysunki, ale mam wrażenie, że zabrakło ich późniejszej obróbki - zadowolono się szybką konwersją kolorów i rozdzielczości do 320x200 i 8 bitów. W efekcie do niektórych lokacji nie chce się wchodzić, a inne wyglądają jak na Amidze z ECS, z charakterystyczną dla trybu HAM rampą. Skoro jesteśmy już przy grafice, to wspomnę jeszcze o dwóch irytujących mnie elementach. Pierwszym z nich jest sposób, w jaki w dużych lokacjach przesuwa się ekran - dzieje się to dopiero po dojściu postaci do jego krawędzi. Drugim natomiast - malutki kursor, mieniający się przy każdym ruchu kolorowymi gwiazdkami. Strasznie mi szkoda, ale za całokształt grafiki należy się ledwo trójczyna.



A oto i doskonale nam znana antropomorficzna personifikacja ludzkich wyobrażeń. Jak zwykle wpada w samą porę.

„Muzyka Duszy”

Co do oprawy muzycznej, to składa się na nią kilka elementów: pierwszym z nich jest muzyka w formacie MIDI (dominująca w uboższej, piętnastodyskiowej wersji), która nie zachwyca i raczej nie zapada w pamięć. Kolejne to dość poprawne efekty dźwiękowe i głosy, o które rozbudowana jest wersja CD. Dzięki nim gra staje się o wiele przyjemniejsza, choć moim zdaniem niektóre głosy można było dobrać trafniej do postaci (do tego te niesynchronizowane ze słowami ruchy ust - gdy przypominam sobie, że „Full Throttle” wydano w tym samym roku, to chce się tylko west-



chnąć). Jako ciekawostkę dodam, że głos Rincewind podkłada znany m. in. z grupy Monty Pythona Eric Idle.

„Teatr Okrucieństwa”

Teraz będzie trochę o grywalności. Nie zamierzam owijać w bawełnę - mnie gra odrzuciła. Nie tylko dlatego, że pod starszymi wersjami ScummVM zawieszala się bez ostrzeżenia. Nie tylko dlatego, że w nowszych stanu gry nie da się zapisać (nie wiem, jak wygląda sprawa na AmigaOS 4.x czy Amidze klasycznej, ale pod MorphOS-em najlepiej grać na ScummVM 1.2.1). Nie tylko z tego powodu, że można utknąć bez wyjścia, gdy w poprzednim akcie nie zdobędziemy jakiegoś przedmiotu. Głównie dlatego, że po lekturze książek spodziewałem się czegoś więcej. Niekoniecznie zrzynki z lektury (gdy przeczytało się książki z cyklu, to mało który zawarty w grze żart jest w stanie wywołać coś więcej niż tylko przelotny uśmiech), koszmarnego poziomu trudności (spróbujcie ukończyć grę bez solucji, no śmiało, śmiało!) i frustrującego wodzenia mieniającym się kolorowymi gwiazdkami kursorem po kolejnych elementach kiepsko pokolorowanej scenerii w poszukiwaniu kolejnego, sprytnie ukrytego, acz kluczowego jej elementu.

„Na Glinianych Nogach”

Podsumowując - jeśli fascynuje Was stworzony przez Terry'ego Pratchetta świat i jesteście zagorzałymi zwolennikami przygodówek, spróbujcie zagłębić się w opisywaną grę. Ostrzegam tylko, że jest ona gigantyczna. Przechodzenie jej - nawet z solucją - może zająć ładnych parę dni. Postarajcie się też o wersję CD - da wam znacznie więcej satysfakcji. Mnie niestety kojarzy się co najwyżej z podejrzaną kielbaską w bułce od „Gardło Sobie Podrynam” Dibblera. Taką grę po prostu należało zrobić przynajmniej o kilka klas lepiej.

Konrad Czuba



Podręczny ekwipaż, czyli „Pokaż chucherko co masz w kuferku”



„Wchodzimy do świątyni Offlera. Chyba nie musimy przypominać, żeby trzymać się wyznaczonej ścieżki zwiedzania?”



Nieznany artysta - „Podmiejski pejzaż z autoportretem”



- oryginalny, Pratchettowski humor
- okazjonalnie - ładna grafika
- przyzwoite udźwiękowanie
- rozległy świat do eksploracji



- obrzydliwe lokacje
- frustrujące sterowanie
- surrealistyczny poziom niektórych zagadek
- błędy w grze potrafiące uniemożliwić jej ukończenie



Czy Amiga może być gwiazdą imprezy?

Miałem kiedyś kumpla, który choć nigdy w życiu nie posiadał Amigi, to za moim pośrednictwem wiedział o niej całkiem sporo. Był również ze mną na kilku amigowych imprezach, jak choćby Amizaduszki. Jego narzeczona nie była „w temacie”, ale jako dziecko grała w „Superfroga”. Co jakiś czas wspólnie organizowaliśmy jakieś spotkania: a to grill, a to jakaś nasiadówka w swoim gronie. Raz nieśmiało zaproponowałem, aby zabrać na taką imprezę Amigę. Kumpel i jego luba zgodzili się. Był to strzał w dziesiątkę. Moja Amiga była okupowana przez zdecydowaną większość spotkania, choć mało kto miał wcześniej styczność z tym komputerem. Amiga z powodzeniem może robić za gwiazdę imprezy, o czym Was postaram się przekonać oraz przedstawić przepis jak tego dokonać. Dodam, że konfiguracja naszej Amigi jest kwestią drugorzędną, choć nie całkowicie obojętną, ale o tym później. Zaczniemy jednak od początku.

Po pierwsze - najpierw trzeba mieć okazję. Co istotne - z dostępem do prądu. Jeżeli nie istnieje tzw. metodą tradycyjną, można sobie jakoś poradzić, np. przy pomocy akumulatora samochodowego i przetwornicy 150 W. Taka przetwornica w zupełności wystarcza, aby zasilic Amigę + monitor „made by Commodore”. Ważne, aby przetwornica dawała cały czas (a nie tylko chwilowo) stałą moc 150 W. Z moich obserwacji wynika, że akumulator 62 Ah, o ile jest całkowicie sprawny i naładowany, wystarcza na kilka godzin zabawy. Mój się nie rozładował. Spokojnie mogłem go włożyć z powrotem do auta i uruchomić silnik.

Po drugie - gdy już mamy okazję, to możemy śmiało wkręcić swój pomysł przytęszczenia Amigi i jej zaprezentowania. Jeśli się uda, to dobra nasza. Jeśli nie, to trudno.

Po trzecie - jeśli punkt drugi wypali, to nie możemy wyjść na głupka. Trzeba mieć świadomość, że większość ludzi kojarzy Amigę z reliktem przeszłości. Dobrze na samej imprezie jest nakreślić aktualną sytuację Amigi i w szczególności polecić PPA. Jednak nie zachęcam do tłumaczenia sytuacji „mosowo-arosowo-asowej”. Raczej nikogo to nie zainteresuje, w szczególności dziewczyn obecnych na imprezie. Niemniej parę faktów można krótko streścić. Bez większego znaczenia jest też to, czy zabierzemy Amigę 500, 600, 1200 czy inną. Na każdej możemy pokazać coś ciekawego i zabawić jednocześnie całe towarzystwo. Grunt, aby zrobić to z głową i klasą. Im lepsza Amiga, tym oczywiście lepiej, ale już na poczywiej „pięćsetce” możemy naprawdę dużo zaprezentować.

Ze swojego doświadczenia wiem, że mało kogo, szczególnie po kilku głębszych, interesują szczegóły techniczne. Na początek polecam odpalanie kilku gier. Bardzo dobrym pomysłem jest zaopatrzyć się w interfejs umożliwiający podłączenie do portu równoległego Amigi dodatkowych „dżojów”. Dzięki temu nawet pięciu graczy będzie mogło na raz pograć przy jednej Amidze. Co ciekawe, bardzo często w gry zręcznościowe wkręcają się także dziewczyny.

Nie raz dochodziło do tego, że była przepychanka, kto teraz ma grać. Jeśli nie możemy takiego interfejsu kupić, to możemy go zrobić samemu. Koszt części to około 13 PLN. Jeśli boimy się lutować, to trzeba kombinować i szukać kogoś, kto nam pomoże. Naprawdę warto.

Zabawa się zaczęła. Ludziska patrzą dużymi oczami na to coś, co przynieśliśmy i pytają „co to za komodor”. Naszym zadaniem jest zaciekać pospólstwo do bliższych oględzin. Możemy w kilku zdaniach powiedzieć, jakie nasza Amiga ma w środku bebechy i z czym to się je. Na sam początek polecam „Dyna Blaster”. Można grać w pięciu graczy, o ile mamy wspomniany wcześniej adapter. Jeśli mamy Amigę 1200 z dyskiem twardym, to proponuję „Blitz Bombers” (do pobrania za darmo z Aminetu). Zdecydowanie lepszy od oryginału klon „Dyna Blaster”. Dalej polecam klasykę: „Skidmarks” lub „Indy Heat” (można grać w trzech). Dziewczyny naprawdę w to grają! W chwili przerwy możemy odpalić jakieś scenowe demo. Oczywiście takie, które na naszej Amidzie odpali. Jeśli mamy Amigę 500, to polecam „Enigmę”. Warto też dodać, z którego jest ono roku.

W co jeszcze warto zagrać? To już zależy od towarzystwa i od posiadanej Amigi. Ilość chętnych oraz zapal są adekwatne do wyboru gry. Na jednej imprezie ludzie grali trzy godziny w „Blitz Bombers” i nie było siły, aby włączyć coś innego. Innym razem zainteresowanych było dwóch panów i tłuśliśmy w „Mortal Kombat” oraz „Super Stardust”. Poza tym, jeśli mamy odpowiednio wyposażoną Amigę, to możemy zaprezentować mocniejsze gry. Mam tu na myśli „Quake” (najlepszy jest standardowy komentarz pospólstwa: „To wyszło na Amigę?”). Poza tym „Earth 2140”, „Duke Nukem 3D”, „Quake II”. Bardzo fajne wrażenie robi emulator Macintosha z uruchomioną grą „Rebel Assault II”. Zresztą - gdy to raz pokazywałem na SACP, to ludziska byli zdziwieni, że to na Amidzie chodzi.

Wróćmy jednak do dziewczyn, których ani emulator Macintosha, ani tym bardziej „Rebel Assault II” nie interesuje. Panie lubią grać w „Superfroga”, co nie dziwi, oraz „Draggy and Croco” czy również masę innych fikuśnych gier, jak choćby „Hugo”. Co ciekawe kobiety lubią też grać w „Mortal Kombat” i parę razy byłem proszony przez dziewczynę o włączenie właśnie tej gry.

Nie samymi grami jednak człowiek żyje. Ja zrobiłem furorę, odpalając na Amidzie 600 z twardym dyskiem kilka utworów muzycznych przerobionych z MP3 do Wave. Dodatkowo puściłem kilka animacji pobranych z Aminetu w formacie CDXL oraz intro z gry „Alien Breed: Tower Assault”. Warto też odpalić intro z gry „Cytadela”. Poza tym na Aminecie jest sporo animacji Erica Schwartza, które można puścić. To naprawdę robi wrażenie na ludziach. Raz nawet przerobiłem teledysk z oryginalnego formatu XVID na 320x240 Radius Cinepak. Powyciągałem pojedyncze klatki, szybkość od-

twarzania ustaliłem na 12 klatek na sekundę (aby Amiga się ze wszystkim wyrobiła). Ścieżkę dźwiękową przerobiłem na IFF-a, a później wszystko pod Scalą poskładałem w animację. Efekt był ciekawy. Ludziska oglądali, a co niektórzy byli bardzo zdziwieni, że Amiga jest w stanie odtwarzać filmy na pełnym ekranie. Do prezentacji użyłem Amigi 1200 z twardym dyskiem i dodatkową pamięcią Fast.

Jest mnóstwo innych fajnych rzeczy, które na Amidzie można podczas takiego spotkania zademonstrować. Ja zrobiłem sobie pióro świetlne. Koszt części to około 6 PLN, a samo wykonanie jest banalne. Czujnik, opornik i możemy szaleć. Wrażenie na widzach robi to spore. Innym razem podłączyłem do Amigi Siemens C35 kupionego za parę PLN na portalu aukcyjnym. Połączenie komórka->komputer uzyskałem za pomocą kabla szeregowego oraz przejściówki z DB25 na DB9. Za pośrednictwem modemu wbudowanego w telefon połączyłem się z internetem i pokazywałem PPA na IBrowse. Efekt równie piorunujący na widzach (artykuł o tym można znaleźć na PPA). Kolejną rzeczą, jaka zrobiła wrażenie, to wykonanie kilku zdjęć aparatem cyfrowym. Akurat mój miał kartę CF. Podłączyłem to od razu do Amigi za pomocą PCMCIA i wyświetliłem zgrane zdjęcie. Tak na marginesie dodam, że warto w aparacie ustawić jak najmniejszą jakość zdjęć, aby nasza Amiga nie dostała czkawki przy wgrzaniu pliku JPEG.

Czasami trzeba się przygotować na różne dziwne pytania - niekiedy bardzo irytujące. Wśród nich chyba koronne „Czym Amiga się różni od komputera”. W miarę spożycia alkoholu tego typu pytania zanikają na rzecz niczym nieskrępowanej i nieorganicznej rozrywki czerpanej z naszego ulubionego komputera. Koniecznie musimy zadbać o bezpieczeństwo naszej Amigi. Mam tu na myśli dwa aspekty: pogodowy i ludzki. Na ten pierwszy musimy być przygotowani, jeśli zabawa jest na świeżym powietrzu albo korzystamy z prowizorycznego zadaszenia. Musimy się liczyć z szybką ewakuacją naszego sprzętu w razie nagłej zmiany pogody. W tym celu przed właściwą zabawą warto jest dogadać się z innym członkiem imprezy i poprosić o ewentualną pomoc. Drugi czynnik jest dużo gorszy i bardziej nieprzewidywalny. Szczególnie gdy ilość spożytych płynów wyskokowych jest adekwatna do czasu trwania imprezy. Trzeba po prostu pilnować, aby nikt nam nie wylał piwa lub innego trunku na komputer. Zdarzyła mi się też sytuacja, że dziewczyna zmieniając dyskietkę w Amidzie wkładała ją na opak. Udało mi się na szczęście odpowiednio wcześniej zareagować.

W sumie to tyle moich rad. Ja swoją Amigę zabieram na większość imprez. Znajomi, gdy mnie zapraszają, to czasami od razu mówią, żebym wziął sprzęt i „dżoje”. Wielu ludzi dzięki temu po raz pierwszy poznało ten fajny komputer albo wróciło na kilka chwil do swoich młodościowych lat. No i wiem też, że co niektórzy zaczęli zaglądać na PPA. Sądzę, że warto wyjść z Amigą do ludu.

Benedykt Dziubałowski



Nasza pierwsza gra

Kurs programowania z wykorzystaniem Visual Studio 2005 Express Edition i VBCC – część 1

Celem tego kursu jest nauka programowania pod AmigaOS z wykorzystaniem Visual Studio 2005 Express Edition i VBCC. Oczywiście ten zestaw można zastąpić innym, bardziej nam odpowiadającym, lecz wówczas kilka pierwszych akapitów należy po prostu pominąć. Jako że nauka najlepiej idzie na przykładach, w dodatku praktycznych, postaramy się stworzyć prostą grę logiczną.

Na wstępie określimy, co będziemy potrzebowali, aby w miarę bezboleśnie tworzyć oprogramowanie dla AmigaOS 2.x/3.x z procesorem mc680x0 na Windowsie używając Visual Studio C++ 2005 Express Edition i VBCC. Prawdopodobnie można zmusić nowsze wersje, czyli 2008 i 2010 do pracy na rzecz Amigi, lecz mój PC jest dosyć leciwy i nawet nie próbowałem instalować nowszych wersji Visual Studio. Będziemy potrzebować wspomniany Visual Studio 2005 C++ Express Edition, VBCC generujący kod dla AmigaOS 2.x/3.x z mc680x0 z poziomu Windows i NDK 3.9 (a w zasadzie tylko dwa katalogi: *include_h* i *include_i*). Dodatkowo, jeśli pokusimy się o kompilowanie narzędzia **Vasm**, to będziemy potrzebowali Windows SDK. Co do Visual Studio C++ 2005 Express Edition to milcząco zakładam, że jest on już zainstalowany. Przejdźmy teraz do VBCC. Mamy dwa wyjścia: na podstawie źródeł skompilować potrzebne narzędzia bądź poszperać w sieci i pobrać je w wersji dla Windowsa. Trzeba zaznaczyć, że wersje znalezione w sieci niekoniecznie muszą być najświeższymi i warto to sprawdzić uruchamiając narzędzia z linii poleceń.

Zajmiemy się kompilacją źródeł na maszynie 32-bitowej. Przede wszystkim tworzymy katalog VBCC na dysku C:\. Oczywiście to tylko przykładowa lokalizacja i można ją sobie dowolnie zmienić, ale trzeba pamiętać o zmianie odwołań wszędzie tam, gdzie to stosowne. Od razu z marszu stwórzmy w katalogu VBCC podkatalogi: *bin*, *config*, *doc*, *targets*. W katalogu *docs* należy umieścić dokumentację do narzędzi, którą można znaleźć w postaci plików PDF na stronie Franka Wille.

VBCC

Po pobraniu źródeł VBCC rozpakowujemy je (najlepiej do jakiegoś katalogu tymczasowego, na przykład C:\tmp). Uruchamiamy Visual Studio 2005 Command Prompt. Przechodzimy do katalogu ze źródłami VBCC i tworzymy katalog *bin* (mamy więc C:\tmp\VBCC\bin) i uruchamiamy:

```
nmake /f Makefile.vs TARGET=m68k
```

Dalej otrzymamy szereg pytań. Na wszystkie pytania typu *y/n* odpowiadamy *y*, a na inne pytania wpisujemy w kolejności: *signed char*, *unsigned char*, *signed short*, *unsigned short*, *signed int*, *unsigned int*, *signed long long*, *unsigned long long*, *float*, *double*. Po kilku chwilach, w zależności od szybkości maszyny, w katalogu *bin* (C:\tmp\VBCC\bin) powinny

Jeśli archiwum ze źródłami VBCC nie zawiera pliku *Makefile.vs*, to nie pozostaje nam nic innego jak stworzenie go na podstawie istniejącego *Makefile*. Na początku usuwamy reguły od linii 20 (*doc/vbcc.pdf*) aż do linii 33 (*bin/osekrm*). Regułę *bin/osekrm* zostawiamy i usuwamy regułę *dist* oraz wszystkie reguły *vcpp*. Następnie zamieniamy wszystkie wystąpienia znaku „/” znakiem „\”. We wszystkich regułach kończących się na *.o* zamieniamy ciąg trzyliterowy „-o” (spacja jest istotna) na „/Fo”. Ponadto zamieniamy wyżej wymienione „-o” na „obj”. Odszukujemy pozostałe reguły, gdzie występuje „-o” i dodajemy „.exe” na koniec reguły i w pliku wynikowym, a ciągi „-o” zamieniamy na „/link /NOLOGO /OUT:”.

znaleźć się pliki: *dtgen.exe*, *VBCCm68k.exe*, *vc.exe*, *vprof.exe*. Kopiujemy wszystkie oprócz *dtgen.exe* do C:\VBCC\bin.

Vasm

Pobieramy i rozpakowujemy tak, jak poprzednio do katalogu tymczasowego (C:\tmp). Uruchamiamy Visual Studio 2005 Command Prompt i przechodzimy do katalogu C:\tmp\vasm. Tworzymy katalog *obj_win32* i uruchamiamy:

```
nmake /f Makefile.win32 CPU=m68k SYNTAX=not
```

Po udanej kompilacji zmieniamy nazwę z *vasmm68k_not_win32.exe* na *vasmm68k_mot.exe* i kopiujemy ten plik do C:\VBCC\bin. Warto dodać, że aby poprawnie skompilować Vasm za pomocą Express Edition, należy posiadać Windows SDK zainstalowany i poprawnie ustawione zmienne środowiskowe INCLUDE i LIB. Najłatwiej jest uruchomić VS2005 Command Prompt i przejść do katalogu, gdzie mamy zainstalowany SDK i uruchomić skrypt SetENV, który ustawi zmienne środowiskowe, a dopiero potem kompilować VBCC.

Vlink

Postępujemy podobnie, jak w poprzednich krokach, czyli pobieramy źródła i rozpakowujemy do katalogu tymczasowego. Uruchamiamy Visual Studio 2005 Command Prompt i przechodzimy do katalogu C:\tmp\vlink. Tworzymy katalog *objects* w C:\tmp\vlink i uruchamiamy:

```
nmake /f Makefile.win32
```

Po zakończeniu kompilacji kopiujemy plik *vlink.exe* do C:\VBCC\bin.

Target

Pobieramy Target archiwum dla AmigaOS 2.x/3.x. Kopiujemy katalog *targets* z zawartością do katalogu C:\VBCC. Tworzymy katalog *config* w C:\VBCC i tworzymy tam plik konfiguracyjny *vc.cfg*. Zawartość pliku konfiguracyjnego znajduje się na stronie 18.

Wszystkie odnośniki do wymienionych archiwów znajdują się w ramce na stronie 19.

Instalacja VBCC

W zasadzie, aby korzystać z dobrodziejstwa skróconej kompilacji, pozostaje nam ustawić zmienne środowiskowe i przekopiować dwa katalogi z NDK. Ustawiamy zmienne, a dla VBCC dajemy C:\VBCC (czyli mamy VBCC=C:\VBCC) i do PATH dodajemy ścieżkę C:\VBCC\bin. Teraz bierzemy się za NDK. Rozpakowujemy archiwum w katalogu tymczasowym C:\tmp kopiujemy katalogi *include_h* i *include_i* wraz z zawartością do C:\VBCC. W ten sposób jesteśmy w stanie sprawdzić czy poprawnie wykonaliśmy instalację VBCC. Uruchamiamy wiersz poleceń i wpisujemy *vc*. Naszym oczom powinna ukazać się informacja „No objects to link”.

VS2005 i pierwszy projekcik

Wybieramy z menu „Tools/Options” i przechodzimy do „Projects and Solutions”. Tam wybieramy „VC++ Directories” i z prawego górnego drop down box zaznaczamy „Include files”. Dodajemy dwa nowe wpisy C:\VBCC\include_h i C:\VBCC\include_i. Dzięki temu będziemy mogli korzystać z „intellisense”, czyli mówiąc kolokwialnie - podpowiadaczki. Wybieramy z górnego menu „File/New/Project” bądź wciskamy CTRL+SHIFT+N. Zaznaczamy „Makefile Project” (rozwijamy „Visual C++” i wybieramy „General”) w „Name” wpisujemy „Boxo” i ustawiamy „Location”. Naszym oczom ukaże się Wizard i w tym przypadku przechodzimy dalej klikając w „Next”. W „Debug Configuration Settings” wpisujemy następujące polecenia.

Dla *Build command line*:

```
nmake /f makefile.mak all
```

Dla *Clean commands*:

```
nmake /f makefile.mak clean
```

Dla *Rebuild command line*:

```
nmake /f makefile.mak clean all
```

Jako „Output” wpisujemy „boxo.exe”. W „Release Configuration Settings” zaznaczamy checkbox „same as debug configuration” i klikamy w „Finish”. Kasujemy filtry „Header files”, „Resource Files”, „Source Files” i plik „readme.txt”. Porzucamy na chwilę VS2005 i przechodzimy do katalogu Boxo. Tam kasujemy „readme.txt” i tworzymy pliki „boxo.c” i „makefile.mak” a w nim umieszczamy zawartość przedstawioną w jednej z ramek na stronie 18.



Plik konfiguracyjny vc.cfg

```
01 -cc=VBCCm68k -quiet %s -o= %s %s -O=%ld -IC:/VBCC/targets/m68k-amigaos/include/ -IC:/VBCC/include_h/
02 -ccv=VBCCm68k %s -o= %s %s -O=%ld -IC:/VBCC/targets/m68k-amigaos/include/ -IC:/VBCC/include_h/
03 -as=vasmm68k_mot -quiet -Fhunk -phxass %s -o %s -IC:/VBCC/include_i/
04 -asv=vasmm68k_mot -Fhunk -phxass %s -o %s -IC:/VBCC/include_i/
05 -rm=del %s
06 -rmv=del %s
07 -ld=vlink -bamigahunk -x -Bstatic -CVBCC -nostdlib -LC:/VBCC/targets/m68k-amigaos/lib/ C:/VBCC/targets/m68k-amigaos/lib/startup.o %s %s
08 -lvc -o %s
09 -ldv=vlink -bamigahunk -x -Bstatic -CVBCC -nostdlib -LC:/VBCC/targets/m68k-amigaos/lib/ C:/VBCC/targets/m68k-amigaos/lib/startup.o %s %s
10 -lvc -o %s
11 -ldv=vlink -bamigahunk -x -Bstatic -CVBCC -nostdlib -LC:/VBCC/targets/m68k-amigaos/lib/ C:/VBCC/targets/m68k-amigaos/lib/startup.o %s %s
12 -ldnodb=-s -Rshort
13 -ul=-l%s
14 -cf=-F%s
15 -ml=500
```

UWAGA! Liczby na początku linii zostały zastosowane tylko w celu wyróżnienia pełnych ciągów w tekście. W pliku konfiguracyjnym każda linia powinna zaczynać się od znaku „myślnika”.

Zawartość pliku „makefile.mak”

```
#
# boxo
#

CC = vc
CFLAGS = -c -c99 -IC:/VBCC/targets/m68k-amigaos/include/ -IC:/VBCC/include_h/

BIN = boxo

LIBS = -LC:/VBCC/targets/m68k-amigaos/lib -lvc -lamiga

LINK = vlink -bamigahunk -Bstatic -CVBCC -nostdlib -s -x c:/VBCC/targets/m68k-amigaos/lib/startup.o

OBJ = boxo.o
LINKOBJ = boxo.o

RM = DEL

.PHONY: all clean

all: $(BIN)

clean:
    $(RM) $(OBJ) $(BIN)

$(BIN): $(OBJ)
    $(LINK) $(LINKOBJ) -o $(BIN) $(LIBS)

boxo.o: boxo.c
    $(CC) $(CFLAGS) boxo.c -o boxo.o
```

Pobieźnie omówię, co z czym się je w tym pliku, a potem zajmijmy się *boxo.c*. Otóż zawiera on reguły, które automatyzują proces kompilacji. Znak „#” oznacza komentarz, czyli dalej za nim aż do końca linii tekst nie będzie brany pod uwagę. Dalej widzimy konstrukcję typu *Nazwa = wartość*. Są to zmienne, dzięki którym nie będziemy musieli za każdym razem wpisywać tych samych rzeczy. Do zmiennej dobieramy się pisząc *\$(nazwa)*. Pierwsza reguła to *.PHONY* - która mówi, że reguły *all* i *clean* nie są plikami. Na wypadek, gdyby ktoś chciał kompilować pliki o takich nazwach. Reguła *all* jest od kompilacji całego projektu. Za pomocą *clean* możemy wykasować obiekty i utworzony plik wykonywalny. *\$(BIN)* nakazuje stworzenie obiektów (plików pośrednich) i zlinkowanie ich. Ostatnie dwie linie określają, w jaki sposób powstaje plik *boxo.o*.

Pierwszy przykładzik zaczniemy klasycznie. Wypiszemy na wyjście „hello”. Oto i kod:

```
#include <stdio.h>

int main(void)
{
    printf("hello\n");
    return 0;
}
```

Budujemy go za pomocą skrótu klawiszowego CTRL+SHIFT+B bądź wybierając „Build->Build Solution” z menu. Jeśli nie popełniliśmy błędów, to w katalogu *boxo* powinniśmy mieć plik *boxo*. Przenosimy go na Amigę i uruchamiając go, możemy podziwiać rezultat.

Po klasycznym przykładzie „hello” rzucamy się w głęboką wodę i napiszemy programik, który pokazuje okienko. Zaczniemy od ogólnego

planu, który będziemy sukcesywnie rozbudowywać. Początkowy plan składa się z trzech bardzo podstawowych kroków. Pierwszy to inicjacja, drugi to główna pętla, a trzeci to zakończenie/zamknięcie. Skupmy się na tym, co chcemy tak naprawdę osiągnąć. Mamy pokazać okienko, czyli musimy je otworzyć, a co za tym idzie trzeba je też później zamknąć. Aby zamknąć nasze okienko, użytkownik musi kliknąć w gadżet zamknięcia, zatem w jakiś sposób musimy czekać na ruch ze strony użytkownika. Nasz plan wygląda teraz tak:

1. Inicjacja
 - otwarcie okna
2. Pętla główna
 - czekanie aż użytkownik kliknie w gadżet zamknięcia okna
3. Zamknięcie
 - zamykamy okno

Zamieńmy plan w kod (po prawej stronie).

Spostrzegawczy czytelnik zauważy, że występują tu, w źródle do języka C, komentarze rodem z C++. Dzieje się to dlatego, że używam standardu C99, który pozwala na takie komentarze, jak również inne użyteczne rzeczy typu deklarowanie zmiennych, gdzie potrzebuję, a nie tylko na początku bloku (jak ma to miejsce w ANSI C). Kilka słów komentarza o kodzie. Zakładamy, że funkcja *init()* zwraca 0 i oznacza to, że wszystko, co związane z inicjacją, zostało pomyślnie wykonane. Z tego względu wykonanie funkcji *loop()* następuje dokładnie wtedy, gdy *init()* zwraca zero. Warto odnotować, że funkcja *close()* zostanie wywołana nawet, gdy *init()* nie zwróci zero. Oczywiście konsekwencją tego będzie sprawdzanie czy okno było otwarte - bo przecież bez sensu zamykać okno, które nie było otwarte.

Aby otworzyć okno, skorzystamy z biblioteki *intuition* i funkcji *OpenWindowTags*. Wynika z tego, że przed otwarciem okna musimy otworzyć bibliotekę *intuition* i sprawdzić czy otwarcie się powiodło, a po zamknięciu okna musimy zamknąć, o ile nie korzystamy w dalszym ciągu z tejże biblioteki. Pozostała nam najtrudniejsza sprawa, czyli pętla główna. Z pomocą przyjdzie nam funkcja *Wait* z biblioteki *exec*. Na szczęście *exec.library* jest zawsze otwarta i nie musimy się martwić sprawdzaniem i zamykaniem jej. Zadaniem *Wait* jest oczekiwanie na zadane sygnały. W wielkim skrócie - sygnały to podstawowy mechanizm odpowiedzialny za komunikację w OS. W każdym razie, jeśli klikniemy w nasze okno w gadżet zamknięcia, to zostanie wysłany odpowiedni sygnał do naszego okna i naszym zadaniem będzie jego właściwa obsługa. W naszym programiku za pomocą *Wait* będziemy czekali między innymi na sygnał z naszego okna. Plan rozrósł się do postaci:

1. Inicjacja
 - otwarcie *intuition*
 - sprawdzenie czy się udało otworzyć, jeśli nie to wychodzimy z programu
 - otwarcie okna
 - sprawdzenie czy okno otwarte, w przeciwnym razie przechodzimy do kroku 3
2. Pętla główna
 - za pomocą *Wait* czekamy na sygnał z okna lub sygnał CTRL+C
 - obsługa sygnałów (jeśli użytkownik kliknął w gadżet zamknięcia, to wychodzimy z pętli)

```
//=====
static int init(void);
static void loop(void);
static void close(void);
//=====
int main(void)
{
    if(0 == init())
    {
        loop();
    }
    close();
    return 0; //ok
}
//=====
static int init()
{
    //otwarcie okna
    return 0; //ok
}
//=====
static void close()
{
    //zamknięcie okna
}
//=====
static void loop()
{
    //czekanie aż użytkownik kliknie
    //w gadżet zamknięcia okna
}
//=====
```



3. Zamknięcie

- jeśli okno było otwarte, to je zamykamy
- jeśli *intuition* było otwarte, to je zamykamy

Jak widać, plan nabrał kolorów. Przedstawie teraz pełny przykład. Programik rozrósł się trochę i z pewnością funkcje *loop()* i *handleWinSignal()* wymagają większego komentarza. W *loop()* czekamy w pętli na dwa sygnały: naciś-

nięcie kombinacji klawiszy CTRL i C lub sygnał z naszego okna. Dalej badamy czy przyszły oczekiwane sygnały i podejmujemy odpowiednie kroki. W przypadku nadejścia CTRL+C ustawiamy flagę wyjścia z pętli i tym samym wychodzimy z *loop()*. Po nadejściu sygnału z naszego okna wywołujemy funkcję *handleWinSignal()*, która zajmie się obsługą komunikatów pochodzących z naszego okna. *HandleWinSignal()* składa się z pętli, w której odbiera-

my wszystkie komunikaty. Za pomocą *GetMsg* odbieramy wiadomości z naszego portu i po przepisaniu interesującej zmiennej odpowiadamy na tenże komunikat. Dopiero po tym badamy, co takiego ciekawego się wydarzyło. Gdy został wciśnięty gadżet zamykania okna, to ustawiamy flagę wyjścia.

Warto uruchomić przykładzik z CLI i poeksperymentować wciskając gadżet zamknięcia okna czy też wciskając CTRL+C w oknie CLI. Jak się można domyśleć, kombinacja CTRL i C nie zadziała w otwartym przez nas oknie, bo nie pochodzi od sygnału z naszego okna. Powyższy przykładzik posłuży nam jako baza do dalszej rozbudowy, bo nie zapominajmy o głównym celu, jakim jest napisanie gry logicznej. Będziemy robić to metodą małych, ale widocznych kroków. Warto w tym miejscu się zatrzymać i podać więcej szczegółów dotyczących naszej logicznej gry.

W dalekiej przyszłości, gdy loty kosmiczne były tak powszechne, jak przepyszne wypiekany chleb z piekarni pana Zenka, a ludzie pouciekali z planety, na której zostały tylko roboty, pojawiło się zagrożenie w postaci braku mechaników konserwujących roboty. Najwyższajniej w świecie inne roboty brzydziły się tą robotą. W tym krytycznym momencie rada najwyższych robotów podjęła wielkie ryzyko i odważyła się wysłać ekipę, która miała na jakimś zagubionym promie kosmicznym odnaleźć ludzi skłonnych podjąć się tego zadania. Ponieważ chętnych nie było i jakoś dziwnym trafem cała populacja robotów akurat skoczyła na kolejkę oleju, zorganizowano wielkie losowanie (coś w stylu totolotka). W ten oto dziwny sposób wybrano do tej arcytrudnej ekspedycji średnio rozgarniętego robota, którego umieszczono w niewielkim statku kosmicznym. W całym zamieszaniu zapomniano wlać większej ilości paliwa do baku pojazdu. Nie było też wzmianki o fatalnym sterowaniu statkiem (tylko w cztery strony). Nie przeszkodziło to jednak naszemu bohaterowi udać się z misją. Niestety poziom inteligencji robota nie pozwalał na poszukiwanie paliwa w kosmosie, lecz na szczęście wynaleziono bardzo ekonomiczny sposób poruszania się: odbijanie się od przeszkód w przestrzeni kosmicznej - czyli zużywanie mikroskopijnej ilości paliwa na nadanie kierunku naszemu statkowi i dalej czekamy, aż w coś uderzymy.

Tak o to pokrótce przedstawia się historia związana z naszą grą. Zadaniem gracza będzie dotarcie do miejsca, gdzie mogą być ludzie. Kierujemy niewielkim statkiem, który może poruszać się w czterech kierunkach i zatrzymuje się dopiero wtedy, gdy uderzy w przeszkodę. Statek ma niewielką wytrzymałość i nie może się odbijać od przeszkód w nieskończoność, a dodatkowo niektóre przeszkody powodują natychmiastową śmierć.

Jak widać, czeka nas sporo pracy nad grą, ale to już w następnym odcinku. Mogę zdradzić, że zajmujemy się *timer.device* i zrobimy porządek w naszym kodzie źródłowym.

Asman

```
#include <proto/exec.h>
#include <proto/intuition.h>
#include <proto/graphics.h>
#include <dos/dos.h>
#include <intuition/intuition.h>
//=====
struct IntuitionBase* IntuitionBase;
struct Window* m_pwin = NULL;
//=====
static int init(void);
static void loop(void);
static void close(void);
static BOOL handleWinSignal();
//=====
int main(void)
{
    if(0 == init())
    {
        loop();
    }
    close();
    return 0; //ok
}
//=====
static int init(void)
{
    IntuitionBase = (struct IntuitionBase*)OpenLibrary("intuition.library", 36L);
    if(NULL == IntuitionBase)
    {
        return -1;
    }
    //=====
    m_pwin = (struct window*) OpenwindowTags(NULL,
        WA_Left, 0,
        WA_Top, 0,
        WA_Width, 320,
        WA_Height, 256,
        WA_CloseGadget, TRUE,
        WA_Title, (ULONG)"boxo",
        WA_Activate, TRUE,
        WA_DragBar, TRUE,
        WA_GimmeZeroZero, TRUE,
        WA_IDCMP, IDCMP_CLOSEWINDOW | IDCMP_RAWKEY,
        TAG_END);
    if(NULL == m_pwin)
    {
        return -1;
    }
    return 0; //ok
}
//=====
static void close(void)
{
    if(m_pwin)
    {
        Closewindow(m_pwin);
    }
    //=====
    if(IntuitionBase)
    {
        CloseLibrary((struct Library*) IntuitionBase);
    }
}
//=====
static void loop(void)
{
    BOOL bEnd = FALSE;
    while(!bEnd)
    {
        ULONG winSignal = 1L << m_pwin->UserPort->mp_SigBit;
        ULONG signals = wait(winSignal | SIGBREAKF_CTRL_C);
        if(signals & SIGBREAKF_CTRL_C)
        {
            bEnd = TRUE;
        }
        if(signals & winSignal)
        {
            bEnd = handleWinSignal();
        }
    }
}
//=====
static BOOL handleWinSignal()
{
    BOOL bEnd = FALSE;
    while(TRUE)
    {
        struct IntuiMessage* pmsg = (struct IntuiMessage*) GetMsg(m_pwin->User-
Port);
        if(NULL == pmsg)
        {
            break;
        }
        ULONG msg_class = pmsg->Class;
        ReplyMsg((struct Message*) pmsg);
        if(IDCMP_CLOSEWINDOW == msg_class)
        {
            bEnd = TRUE;
        }
    }
    return bEnd;
}
//=====
```

Odnosiniki:

- <http://www.ibaug.de/VBCC/amiga/VBCCwin.zip> – VBCC w wersji dla Windows
- <http://www.ibaug.de/VBCC/VBCC.tar.gz> – źródła VBCC
- <http://sun.hasenbraten.de/vasm/daily/vasm.tar.gz> – źródła Vasm
- <http://sun.hasenbraten.de/vlink/daily/vlink.tar.gz> – źródła Vlink
- http://mail.pb-owl.de/~frank/VBCC/2011-08-05/VBCC_bin_amigaos68k.lha – archiwum Target dla AmigaOS 2.x/3.x



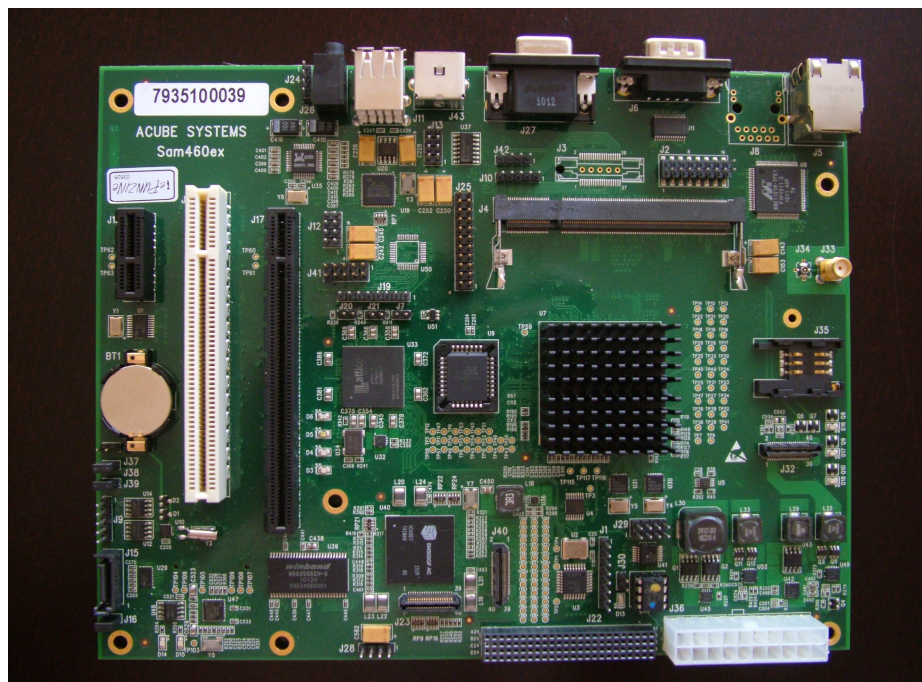
Sam460ex

W chwili, gdy po latach oczekiwania pojawił się w sprzedaży AmigaOS 4.0, płyta Amiga-One nie była już produkowana. To postawiło sympatyków AmigaOS w dość kłopotliwej sytuacji. Na szczęście, na rynek amigowy odważyła się wejść firma ACube Systems, która powstała w wyniku połączenia kilku mniejszych włoskich spółek o amigowych korzeniach. Pod koniec roku 2008 wyprodukowana została pierwsza od czasów AmigaOne, dedykowana dla AmigaOS płyta główna - Sam440ep. Niedługo później przygotowano bardziej rozszerzalny model tej płyty - Sam440ep-flex. Płytom serii Sam zarzucano niską wydajność w stosunku do ceny, co jednak nie zniechęciło producenta, który nie spoczął na laurach. W dniu 27 stycznia 2011 roku firma ACube Systems wprowadziła do sprzedaży najnowszy model płyty pod nazwą Sam460ex.

Sprzęt

Serce nowej Sam - AMCC PowerPC 460ex, taktowany zegarem 1.15 GHz - jest układem typu „System-on-Chip” przeznaczonym dla rozwiązań wbudowanych. Co najważniejsze, zawiera on wiele technologii, które do tej pory na Amidze nie były dostępne - w szczególności kontroler PCI Express oraz SATA II. W porównaniu do modelu 440ep jest to oczywisty krok naprzód. Rdzeń procesora zastosowanego w 460ex jest bardzo podobny do tego znanego z 440ep (gdyż jest jego ewolucją), więc zaadaptowanie AmigaOS do działania na tym procesorze prawdopodobnie było dużo łatwiejsze, niż gdyby skorzystano z procesora innej rodziny.

Sam460ex daje możliwość zainstalowania do 2 GB pamięci RAM DDR2 w slotach SODIMM, co wydaje się sporą ilością jak na dzisiejsze, amigowe wymagania. Pamięć jest taktowana częstotliwością, która jest zależna od zainstalowanego modułu (maksymalnie 575 MHz, dlatego zalecane jest użycie modułów 667 lub 800 MHz).



Grafika obsługiwana jest przez układ Silicon Motion SM502 oraz 64 MB pamięci DDR. Firma Silicon Motion znana jest głównie na rynku rozwiązań wbudowanych. Układ ten posiada kilka istotnych ograniczeń. Maksymalna obsługiwana rozdzielczość to 1280x1024 (proporcje 4:3) lub 1440x900 (16:9). Ponadto, nie obsługuje on akceleracji 3D w żadnej formie. Kłopotliwy jest też brak wyjścia DVI. Płyta główna Sam460ex wyposażona jest jedynie w port VGA oraz złącze LVDS (na płycie). Użytkownicy, dla których kwestie te mają znaczenie, będą musieli zainwestować w osobną kartę graficzną Radeon.

Za dźwięk odpowiadają układy SM502 (kontroler AC'97) oraz Realtek ALC655 (kodek AC'97). Na tylnym panelu znajdziemy tylko jedno gniazdo - wyjście dźwięku. Pozostałe wyprowadzenia kodeka (wejście liniowe, mikrofon) znajdują się na złączu AC'97 na płycie. Pomimo zastosowania kodeka 5.1, użytkownik ma do dyspozycji jedynie wyprowadzenia stereofoniczne. Wszystkie wejścia i wyjścia dźwięku są wyłącznie analogowe - cyfrowa transmisja nie jest obsługiwana.

Karty rozszerzeń można instalować w trzech slotach: PCI-Express 16x (działa w trybie 4x), PCI-Express 1x oraz PCI. Slot PCI obsługuje karty taktowane 33 i 66 MHz, zasilane napięciem 3.3V. Próba instalacji karty 5V może spowodować uszkodzenie.

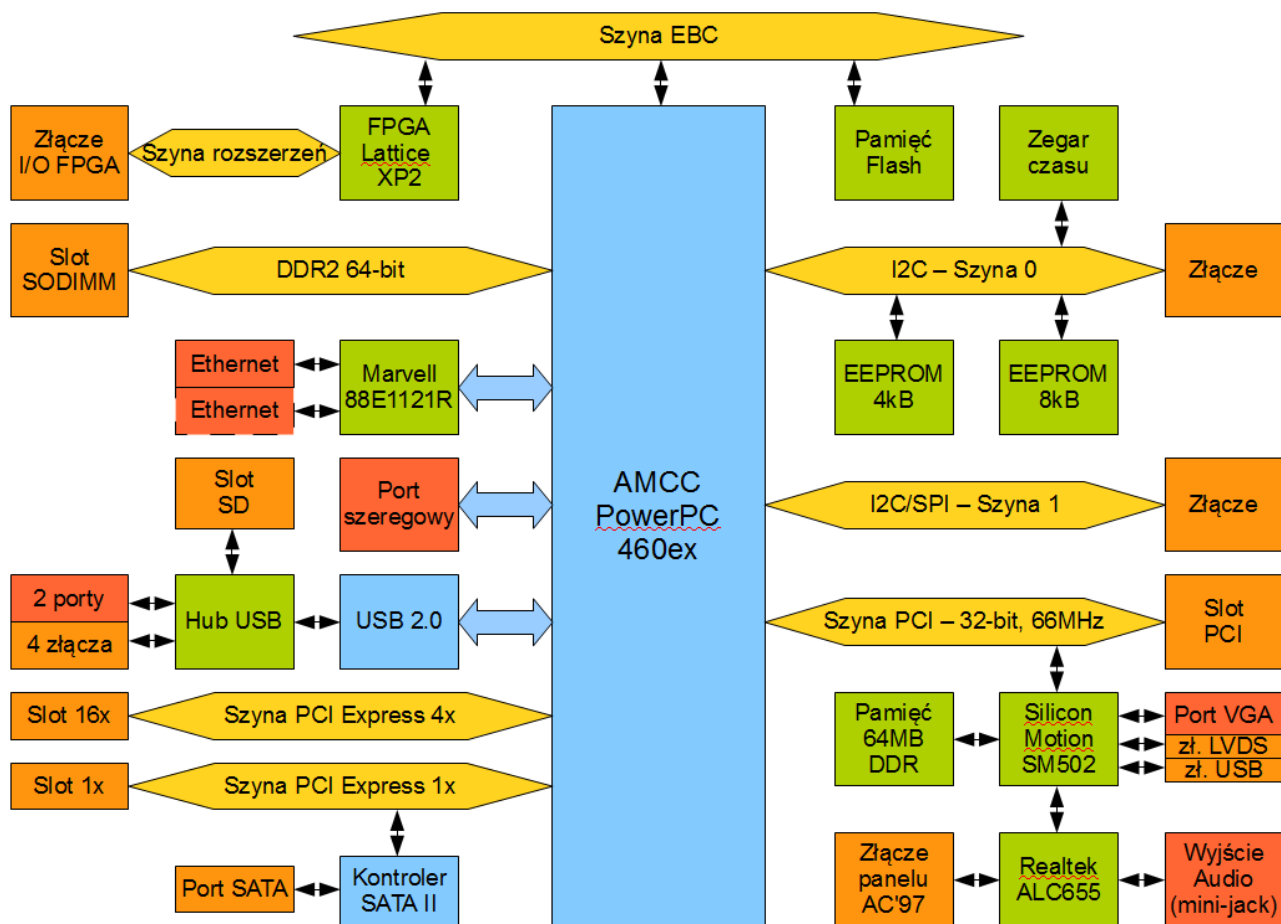
Na płycie znajduje się jeden port SATA II. Często pada pytanie, dlaczego w płytach głównych Sam440ep znajdowały się aż cztery porty, a nowsza Sam460ex ma tylko jeden. Odpowiedź nasuwa się sama, gdy spojrzymy na specyfikację procesorów - model 440ep nie posiadał w ogóle wbudowanego kontrolera SATA i na płycie umieszczono układ Silicon Image 3114. Skoro 460ex posiada wbudowany jednokanałowy kontroler SATA, umieszczanie dodatko-

wego kontrolera na płycie zbędnie podniosłoby koszty - część użytkowników w ogóle nie korzysta z napędów CD/DVD, więc posiadanie jednego kanału SATA wydaje się w takiej konstrukcji wystarczające. Z kontrolerem SATA związana jest jeszcze jedna niedogodność - nie ma możliwości jednoczesnego wykorzystania slotu PCI Express 1x oraz wbudowanego kontrolera SATA. To także wynika z konstrukcji układu zastosowanego w 460ex. Dlatego niektórzy użytkownicy ignorują istnienie wbudowanego kontrolera i instalują kartę SATA w slotcie PCI-Express 1x.

Sam460ex posiada aż sześć portów USB 2.0, z których dwa dostępne są na tylnym panelu. Kontroler EHCI jest zintegrowany w procesorze. Dodatkowo użytkownik ma do dyspozycji jeden port USB 1.1 (obsługiwany przez SM502).

Podziękuję budzącą kontrowersję jest układ FPGA Lattice XP2, o rozmiarze 5000 LUT (ang. „Look-Up Tables”). Wiele osób podaje w wątpliwość sens umieszczania takiego układu na płycie przeznaczonej dla końcowego użytkownika. Z układu FPGA wyprowadzone jest na złącze I/O 80 połączeń, w oparciu o które można budować własne rozszerzenia. Jeśli znajdzie się ktoś na tyle odważny, aby eksperymentować z podłączaniem ich do Sam, czemu nie...

Na spodzie płyty znajdziemy także slot kart SD/SDHC, który można wykorzystać do wymiany danych oraz z którego można uruchamiać system operacyjny (jest on zwykłym urządzeniem pamięci masowej na szynie USB). Jedynie jego umiejscowienie pozostawia nieco do życzenia - dostęp do slotu jest co najmniej kłopotliwy, gdy płyta zainstalowana jest w obudowie typu „wieża”.



Schemat blokowy Sam460ex.

Najnowsza Sam posiada również szereg złączy, które można wykorzystać w rozwiązaniach wbudowanych, a które dla amigowców nie mają obecnie szczególnego znaczenia: slot na kartę SIM, złącza I2C/SPI, JTAG, złącze modułu UMTS/GSM, możliwość instalacji pamięci NAND flash itd. Sugeruje to, że firma ACube ma też innych (poza amigowcami) klientów.

Zasilanie doprowadzane jest poprzez 20-pinowe złącze ATX.

Na tylnym panelu znajdują się następujące porty:

- port Ethernet;
- port szeregowy RS-232C;
- wyjście VGA;
- port S-Video (nieaktywny, wymaga dodatkowego modułu);
- 2 porty USB;
- wyjście dźwięku (wtyk minijack).

Opcjonalnie mogą się tam znaleźć też: drugi port Ethernet oraz wyjście anteny modułu GSM.

Instalacja sprzętu

Produkt jest dostarczany w niewielkim, kartonowym pudełku z logo „Sam460ex - power for your embedded ideas”. Zawiera ono porządknie zapakowaną płytę główną, system operacyjny AmigaOS 4.1 Update 2, blaszkę portów do obudowy, instrukcję obsługi do płyty głównej, podręcznik instalacji AmigaOS dla Sam460ex oraz książeczkę „szybki start” z AmigaOS 4.1. Naturalnie, wszystkie dokumentacje są w języku angielskim. Instrukcja do płyty zawiera nie-

zbędne minimum informacji (np. jak podłączyć przewody panelu frontowego, opis wyprowadzeń złączy), ale mogłaby być znacznie dłuższa. Brakuje w niej np. dokładniejszej listy kompatybilnego sprzętu, z którym może działać obecnie Sam460ex (jest co prawda lista kontrolerów, z których może uruchamiać system) oraz opisu najważniejszych komend firmware.

Sam460ex jest płytą standardu Flex ATX, ma wymiary 21.6 cm x 17 cm. Skoro rozmiar jest niewielki, a płyta nie wymaga aktywnego chłodzenia, można pokusić się o zbudowanie bezgłośnego systemu. Do zasilania płyty można wykorzystać nawet zasilacz typu DC-DC na 12V (ma takowe w ofercie np. firma Mini-Box). Ja zdecydowałem się wykorzystać klasyczną obudowę typu „wieża”, a konkretnie Fractal Design Arc Mini. Osoby uważnie śledzące projekt AmigaOne X1000 zauważą, że Arc Mini jest mniejszą siostrą obudowy Define, która oferowana ma być w komplecie z X1000. Zastanawiające jest, dlaczego dystrybutorzy nie wpadli na pomysł sprzedawania Sam460ex z obudową Arc Mini, której rozmiar jest odpowiedni, a stylizyka idealnie nawiązuje do bardziej prestiżowego projektu X1000. Zwykle, tanie obudowy PC oferowane obecnie do serii Sam przez firmy takie jak AmigaKit, Relec, a nawet ACube, nie robią pozytywnego wrażenia. Dlaczego oszczędzać na obudowie, gdy kupuje się tak kosztowny sprzęt?

Tym razem nie popełniono błędu, który miał miejsce w przypadku modelu 440ep. Nowa Sam dostarczana jest z solidną blaszką portów, która wygląda bardzo elegancko. Jej jedyną wadą jest to, że bardzo łatwo się brudzi i

widąc na niej każdy odcisk palca.

Na temat samego procesu budowania komputera, w oparciu o Sam460ex, nie da się napisać zbyt wiele. Jeżeli ktoś kiedyś „składał” komputer PC, to nie odkryje tu niczego rewolucyjnego. W końcu płyty serii Sam projektowane są do współpracy z typowymi komponentami PC.

Podczas testowania zestawu komputer wiele razy jest wyłączany i włączany, aby sprawdzić poprawność połączeń oraz konfiguracji. Na początku byłem zdziwiony, że komputer nie chce ponownie „odpalić” zaraz po wyłączeniu. Okazało się, że Sam460ex ma wbudowane zabezpieczenie przed zbyt szybkim ponownym włączeniem. Zalecane jest oczekiwanie 15 sekund przed ponownym włączeniem zasilania. Jest nawet na ten temat wzmianka w instrukcji, którą pominąłem przy pierwszym jej czytaniu. Przycisk włączenia zasilania także działa nieco inaczej, niż w PC - jego krótkie przyciśnięcie powoduje reset komputera (na początku jest to irytujące).

Oprogramowanie

Tak jak w przypadku AmigaOne oraz Sam440ep, w najnowszej odmianie Sam zastosowano firmware U-Boot. Obsługuje on uruchamianie systemu operacyjnego z napędów CD/DVD, dysków twardych podłączonych do wewnętrznego kontrolera SATA II, kontrolerów na szynie PCI/PCI-Express (Silicon Image 3112, 3114, 3512) oraz urządzeń pamięci masowej USB (w tym wbudowanego slotu kart SD). Aktualizacji firmware można dokonywać



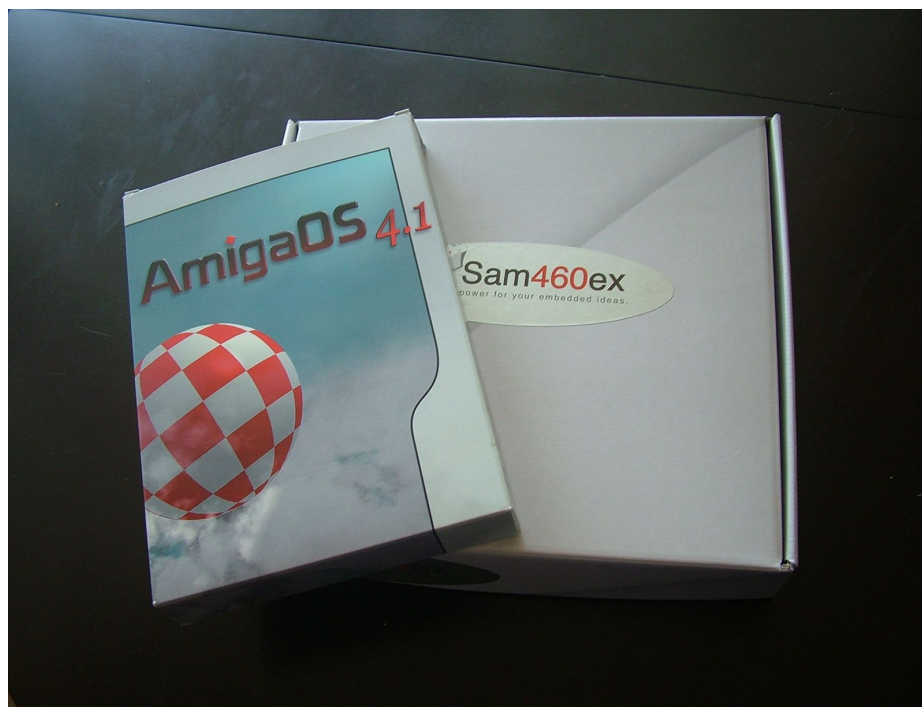
po sieci za pomocą protokołu TFTP lub używając napędu CD - od czasu wydania modelu 460ex pojawiła się już jedna taka aktualizacja. Kod źródłowy całego firmware jest dostępny na licencji GPL (do pobrania na stronie Acube Systems).

Instalacja AmigaOS 4.1 okazała się o wiele bardziej skomplikowana niż sądziłem. Zakładałem, że kupując Sam460ex „zwyčajnie” będę w stanie zainstalować na niej system. Niestety, po podłączeniu do wbudowanego portu SATA II dysku twardego okazało się, że nie miałem gdzie podłączyć napędu CD, a przecież system operacyjny dostarczany jest wyłącznie na takim nośniku. W pierwszej kolejności pomyślałem o wykorzystaniu wbudowanego slotu kart SD. ACube Systems oferuje użytkownikom wsparcie, więc zadałem pytanie, czy możliwe jest przygotowanie karty SD z obrazem instalacyjnym AmigaOS 4.1 na AmigaOS 3.x (lub innym systemie). Otrzymałem odpowiedź, że przygotowywanie kart SD jest wspierane jedynie na AmigaOS 4.1, czyli trzeba już posiadać jakąś działającą maszynę z tym systemem. Ja akurat nie posiadałem.

W kolejnym kroku pomyślałem o wykorzystaniu USB. Teoretycznie istnieje możliwość uruchomienia systemu z płyty CD w napędzie podłączonym przez port USB. Jednak ja nie miałem pod ręką żadnego mostka USB-SATA, a znikoma wydajność USB (maksymalnie 12 Mbit/s) w AmigaOS 4.1 Update 2 zniechęcała mnie do jego szukania.

Szczęśliwie okazało się, że posiadam kompaktowy kontroler Silicon Image 3114. Jego wykorzystanie okazało się właściwą drogą. Początkowo dysk twardy pozostawiłem podłączony do wewnętrznego kontrolera SATA II, a napęd CD podłączyłem do kontrolera PCI. W tej konfiguracji nie udało mi się zainstalować systemu na dysk twardy. Sterownik, z nieznanymi mi powodów, był aktywny dopiero po zainstalowaniu AmigaOS na dysk, dlatego oba urządzenia musiały być podłączone do kontrolera SiI3114 (po instalacji kabelek Serial ATA mogłem już przelożyć do portu na Sam).

Podsumowując, jeśli nie posiadamy innej ma-



szyny z AmigaOS 4.x, to najprostszą i najszybszą opcją jest dołożenie kontrolera SATA na PCI - przynajmniej na czas instalacji. Niektórzy dystrybutorzy, za niewielką opłatą, mogą przygotować kartę SD, co też wydaje się sensownym rozwiązaniem.

Proces instalacji systemu nie był szczególnie intuicyjny, ale udało mi się przebrnąć przez niego w kilkadziesiąt minut.

Pierwsze wrażenie po zainstalowaniu było bardzo pozytywne, szczególnie zwracała uwagę dobra wydajność (w porównaniu do AmigaOS 4 na innych maszynach). Mimo tego, że producent ostrzega, iż dostarczana wraz z płytą wersja systemu powinna być traktowana jako beta, nie odnotowałem problemów ze stabilnością systemu.

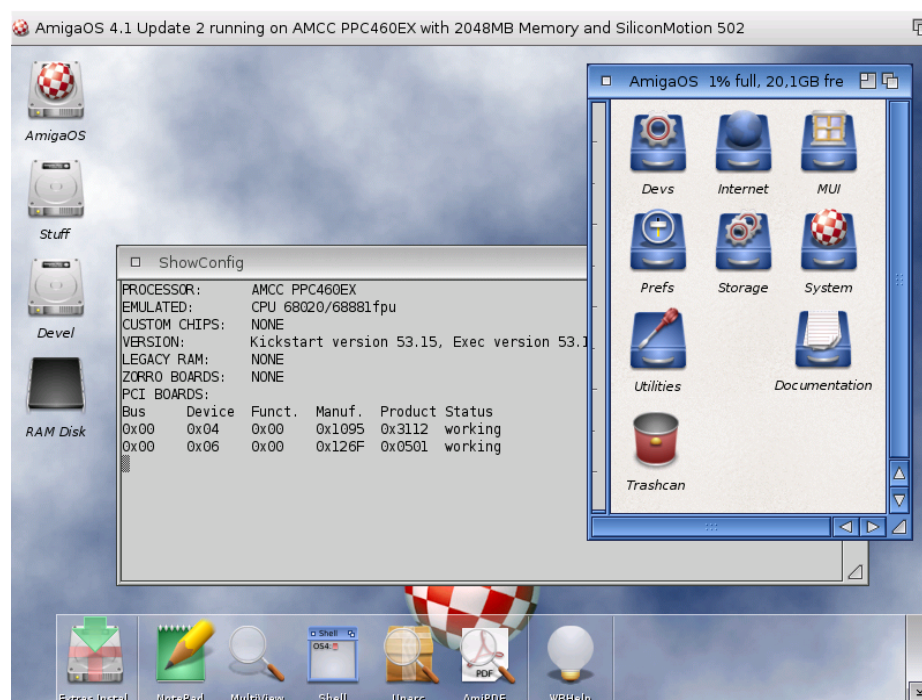
Przez kilka pierwszych miesięcy wsparcie dla nowego sprzętu było mierne. Na szczęście stopniowo pojawiały się sterowniki oraz aktualizacje. Najważniejszą była wydana 28 sierpnia 2011 roku trzecia aktualizacja AmigaOS 4.1. To wydanie przyniosło zmiany istotne dla użytkowników Sam460ex:

- uaktualnione sterowniki SATA, szczególnie sterownik SATA II w układzie PPC 460ex;
- sterowniki USB 2.0, obsługujące kontrolery standardu EHCI, w tym wbudowany w PPC 460ex;
- obsługę szyny I2C;
- sterownik FPGA, pozwalający na oprogramowywanie układu FPGA i wykorzystanie 80-pinowego złącza I/O;
- nową wersję sterownika grafiki SM502;
- sterownik wbudowanej karty dźwiękowej SM502/ALC655 (teoretycznie, ale o tym za chwilę).

Mimo wydania aktualizacji pewne problemy dalej nie zostały rozwiązane. W dalszym ciągu brak wsparcia dla akceleracji 3D na kartach zainstalowanych w slotcie PCI-Express (obecnie prowadzone są prace nad sterownikiem 3D dla kart Radeon HD). Ze strony firmy Hyperion pobrać możemy jedynie archiwum (do doinstalowania na AmigaOS 4.1 Update 2), a nie pełny obraz ISO, co powoduje, że problemy z procesem instalacji dalej mają miejsce. Chyba że użytkownik sam przygotowuje specjalny obraz ISO albo kartę SD do tego celu.

W dokumentacji dołączonej do aktualizacji widnieje informacja, iż zawiera ona sterownik dźwięku dla Sam460ex. Po dokładnym przeszukaniu archiwum stwierdziłem, że niczego takiego tam nie ma, a sterownik należy pobrać ze strony OS4Depot. Sterownik dźwięku w wersji beta był tam dostępny już na kilka dni przed wydaniem uaktualnienia.

Niestety, po zainstalowaniu aktualizacji pojawił się też nowy problem - moja klawiatura nagle przestała działać. Okazało się, że od tej pory muszę ją podłączyć bezpośrednio do portu USB w Sam, a nie poprzez zewnętrzny hub.





Liczyłem też na większe poprawki w sterowniku SM502. Maksymalną głębokością obrazu obsługiwaną przez ten sterownik jest 16-bit, choć sam układ obsługuje też 32-bity. Niestety jego wydajność także nie jest zachwycająca (co zauważyć można na przykład przesuwając po ekranie okno OWB, gdy system jest nieco obciążony).

Jedynym systemem poza AmigaOS, wspieranym na Sam460ex jest Linux. Jeżeli chcemy go używać jako drugiego systemu, zalecana jest wymiana bootloadera SLB na alternatywny Parthenope (ub2lb), napisany przez Giuseppe Coviello z zespołu CruxPPC. Wersja 2.7a tej dystrybucji, wydana w lipcu ma wsparcie dla Sam460ex. Parthenope jest dostępny wraz z kodem źródłowym na stronie ACube Systems.

Wydajność

Z pewnością wielu użytkowników zastanawia się czy warto wymienić Sam440ep na model 460ex ze względów wydajnościowych. W tej kwestii starsza płyta pozostawiała wiele do życzenia. W porównaniu do procesora 440ep, odnotować można drastyczny wzrost wydajności, spowodowany dodaniem 256 kB pamięci cache drugiego poziomu oraz znacznie wyższą częstotliwością taktowania. Dodatkowo układy wykorzystane w Sam460ex są fabrycznie podkręcone z 1.0 GHz na 1.15 GHz. Dzięki temu zwiększane są też częstotliwości taktowania szyny procesora oraz pamięci. Fabrycznie procesor 460ex może pracować z pamięcią taktowaną do 400 MHz, jednak po przetaktowaniu procesora częstotliwość ta może wynosić do 575 Mhz.

Do zmierzenia wydajności procesora wykorzystano program **nbench**. Wyniki testów przedstawia poniższa tabela. Zainteresowanym porównaniem wydajności procesora z innymi maszynami polecam odwiedzenie strony <http://amigadev.free.fr/powerpc/nbench.html>

	Sam460ex 1.15 GHz	Sam440ep 667 MHz
NUMERIC SORT	519.47	311.67
STRING SORT	18.394	10.622
BITFIELD	1.8797e +08	1.0995e +08
FP EMULATION	57.901	33.59
FOURIER	7163.2	4264.3
ASSIGNMENT	7.4683	3.2279
IDEA	1638.7	960.28
HUFFMAN	868.06	507.86
NEURAL NET	8.1967	4.8575
LU DECOMP	159.75	77.92



Tabela poniżej przedstawia wydajność pamięci RAM. Test wykonano programem **RageMem** z pamięcią taktowaną 460 MHz. Z innym modulem pamięci, pracującym przy 575 MHz, wyniki mogłyby być jeszcze lepsze.

READ32	278 MB/Sec
READ64	278 MB/Sec
WRITE32	452 MB/Sec
WRITE64	451 MB/Sec
WRITE	836 MB/Sec (Tricky)

Start komputera z AmigaOS 4.1, od włączenia zasilania do gotowego do pracy Workbench, trwa około 40 sekund. „Miękki” reset za pomocą kombinacji klawiszy Ctrl-Amiga-Amiga trwa około 18 sekund. Wartości te zmierzone były przy wykorzystaniu dysku podłączonego do kontrolera Silicon Image 3114 w (domyślnym) trybie PIO, więc warunki testu nie były optymalne.

Wydajność sieci także nie jest zła, choć można było oczekiwać lepszych wyników. Podczas transmisji dużego pliku (581 MB) protokołem FTP udało mi się uzyskać średnią przepustowość 11 MB/s. Podczas testu sieć pracowała z przepustowością 1 Gbit/s, a obie maszyny testowe były podłączone do tego samego przełącznika Ethernet.

Firma ACube Systems obiecuje, że kolejne aktualizacje firmware i sterowników przyniosą dalsze zwiększenie wydajności. Z pewnością Sam460ex będzie jeszcze wspierana przez długi czas. Wystarczy spojrzeć na Sam440ep – minęło już kilka lat od wypuszczenia jej na rynek, a nadal pojawiają się aktualizacje i usprawnienia dla tej płyty.

Podsumowanie

Sam460ex to obecnie najnowocześniejszy sprzęt, na którym można uruchomić AmigaOS 4.1. Tak zapewne pozostanie do czasu rozpoczęcia sprzedaży komputera AmigaOne X1000. Co prawda jej graficzne i dźwiękowe możliwości są bardzo podstawowe (jak na dzisiejsze realia), ale użytkownikom o większych wymaganiach, płyta oferuje solidną podstawę do zainstalowania lepszej karty graficznej czy dźwiękowej.

Przez pewien czas istotną kwestią był brak sterowników do części wbudowanych urządzeń, ale od wydania AmigaOS 4.1 Update 3 jedynym poważnym problemem pozostaje brak wsparcia dla akceleracji 3D. Zapewne z zakupem wstrzymają się użytkownicy, dla których ta funkcjonalność jest kluczowa.

Na pytanie „czy warto kupić Sam460ex” każdy musi odpowiedzieć sobie sam. Ja uważam, że mimo pewnych wad, jest to bardzo udany produkt. Cena nie jest niska, ale trzeba brać pod uwagę, że jest dużo niższa niż przewidywana obecnie cena AmigaOne X1000.

W Polsce Sam460ex wraz z AmigaOS 4.1 można nabyć w firmie eFUNZINE, w cenie 3699 PLN. Płyta objęta jest gwarancją na okres 24 miesięcy. Bezpośrednio u producenta można nabyć wersję „Lite”, z wolniejszym procesorem 1 GHz oraz 512 MB RAM w cenie 599 euro (w cenę także wliczony jest koszt AmigaOS 4.1 - oferta ważna do wyczerpania zapasów).

Radosław „Strim” Kujawa

Odnosiniki

- <http://www.acube-systems.biz/> – strona ACube Systems, producenta płyty głównej Sam460ex
- <http://www.efunzine.com/> – strona eFUNZINE, dystrybutora płyty głównej na Polskę
- <http://www.hyperion-entertainment.biz/> – strona Hyperion Entertainment, dostawcy AmigaOS 4.1



PyMUI

czyli tworzymy okna Pythonem

Z rozmów z początkującymi użytkownikami MorphOS-a wiem, że dużo osób, próbując pisać swoje pierwsze aplikacje, zraziło się do języka C i drzewiastych struktur MUI. Jeżeli jesteś jedną z nich, mam dla Ciebie dobrą wiadomość - z PyMUI pokochasz tworzenie okienkowych aplikacji. A co to jest? Jest to pakiet dla języka Python, który pozwala tworzyć interfejsy użytkownika przy pomocy środowiska MUI. Z samym MUI każdy, nawet początkujący amigowiec, powinien spotkać się wielokrotnie. Jeżeli jednak słyszysz o MUI pierwszy raz, polecam odwiedzić stronę autora i przeczytać przynajmniej opis tego interfejsu użytkownika. Co do samego Pythona instalację i lekcję wprowadzającą zrobił Marek Hać w poprzednim numerze naszego magazynu, a więc ten temat skrzętnie pominę.

Dlaczego w Pythonie, a nie w C?

Oprócz prostoty tworzenia okien jest jeszcze parę innych aspektów, w których Python przeważa nad C. Dzięki nim polubisz i szybko przyswoisz pracę z Pythonem. Oto kilka z nich:

- kod Pythona jest bardziej czytelny niż w przypadku C - nie będziesz musiał zagłębiać się i szukać gdzie się kończy funkcja i czy nie zapomniałeś zamknąć nawiasów;
- już nie będziesz musiał rekompilować kodu po jego zmianie - po prostu uruchom program jeszcze raz;
- będziesz miał dostęp do nowoczesnego języka obiektowego - swoje programy po drobnych zmianach, głównie w funkcjach wyświetlania interfejsu graficznego, uruchomisz na każdym systemie, dla którego istnieje port Pythona 2.5.x i wyżej.

Jest jeden mały minus - program jest trochę wolniejszy niż jego odpowiednik napisany w C, ale przy obecnych prędkościach procesorów PPC w PowerMacach i Macach Mini nie stanowi to dużego problemu.

Zaczynamy

Zgodnie ze wstępem zakładam, że mamy już zainstalowanego Pythona przynajmniej w wersji 2.5.4. Pobieramy pakiet PyMUI do ze strony autora, rozpakowujemy i instalujemy (oczywiście nazwa archiwum może być inna) komendą:

```
XADUnFile RAM:PyMUI-0.4.morphos.1ha SYS:
```

Po tej operacji mamy już w pełni przygotowaną platformę do pracy. Teraz powinniśmy mieć już działającą wersję PyMUI w systemie, co sprawdzimy w shellu Pythona wpisując (uwaga na gwiazdkę):

```
python> from pymui import *
```

Jeżeli interpreter Pythona nie wyrzucił nam żadnego błędu, to znaczy, że wszystko mamy w porządku i możemy przejść do następnego

roku, którym jest oczywiście prosta aplikacja „Hello World”. Nie zapomnijmy tutaj o użyciu edytora. Szkoda byłoby nie zachować efektów naszej pracy. Edycji możemy dokonać przy pomocy „Eda” z pakietu developer dla MorphOS-a. Ważne tylko by w konfiguracji zaznaczyć, żeby edytor zamieniał tabulatory na spacje. W innym wypadku dość często interpreter Pythona pokaże nam błędy składni.

Hello world

Będzie to przykład zupełnie prosty, zaczniemy od stworzenia okna z przyciskiem oraz notyfikacją, gdzie kliknięcie w obiekt „SimpleButton” spowoduje wyjście z programu. Teraz moduł PyMUI zaimportujemy samym poleceniem `import`. Taki sposób pozwoli nam na łatwe rozróżnienie metod i zaimportowanych pakietów. Przy pisaniu większych programów możemy się spotkać, że w dwóch pakietach występuje metoda o tej samej nazwie. Taki problem, możemy rozwiązać poprzez importowanie bez polecenia `from` lub zaimportowanie tylko określonych modułów z pakietu.

Ciekawym elementem kodu jest sekcja przypięcia notyfikacji, gdzie do wywołania funkcji używamy wyrażenia `lambda`. Wyrażenie to pozwala zdefiniować jednoliniową funkcję. Dzięki jej prostocie nie musimy używać ani nazwy funkcji, ani zwracać wyniku przez słowo kluczowe `return`. `Lambda` pozwala na oszczędzenie dużej ilości niepotrzebnego kodu. Zapamiętajmy - tam, gdzie funkcja jest trywialna i używamy jej w tym konkretnym miejscu kodu, używamy wyrażenia `lambda`. A więc do klawiatu-

wyświetli wiadomości z podanej przez użytkownika strony. Zaczniemy go pisać od zaimportowania modułów `xml.dom.minidom` (pozwoli nam parsować dokument pobrany z sieci), `urllib` (pobierzemy plik ze strony www) oraz `pymui` (do wyświetlenia interfejsu graficznego). Sam kod aplikacji rozpoczniemy od stworzenia grupy obiektów wywołaniem klasy `pymui.Group()`, obiektów `pymui.Text()` pozwalających wyświetlać informacje tekstowe, a także dodamy do formularza pole wejściowe `pymui.String()` i prosty przycisk `pymui.SimpleButton()`. Po stworzeniu obiektów musimy wszystkie elementy dodać do grupy, co czynimy metodą `AddChild(obiekt)`. Kolejne obiekty MUI dodadzą się do naszej grupy w kolejności ich wywołań. Do przycisku dodamy teraz notyfikację, która po kliknięciu w niego wywoła funkcję `pobierzRss()`, do której prześlemy wartość wprowadzoną w polu `url_rss`. Następnie dodamy funkcję, która parsuje nasz dokument, wyszukując nazwy i odnośniki do artykułu w pobranym dokumencie. Tak pobrane dane dodamy do obiektu lista i pozwolimy wyświetlić w aplikacji. A więc bierzmy się do pisania (listing na stronie 25).

Co to jest PyMUI?

PyMUI to pakiet dla skryptowego języka Python. Zastosowanie znajdzie w Pythonie w wersji 2.5.4 lub nowszej. Jego zadaniem jest stworzenie aplikacji w języku skryptowym, która do zbudowania graficznego interfejsu użytkownika wykorzysta MUI (Magic User Interface).

```
#!python
# Importujemy pakiet pymui
import pymui

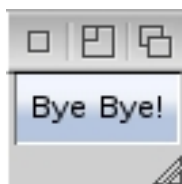
# Tworzymy obiekt Aplikacji, okno i przycisk
aplikacja = pymui.Application(Base='PyMUI_HelloWorld', Title='Hello world', Description='Prosty przykład')
okno = pymui.Window('Hello, world!')
przycisk = pymui.SimpleButton('Bye Bye!')

# Przypisujemy okno do aplikacji i ustawiamy jako główny obiekt przycisk
aplikacja.AddChild(okno)
okno.RootObject = przycisk

# Ustawiamy notyfikację, która w momencie naciśnięcia przycisku bądź znaku x wychodzi z programu
okno.Notify('CloseRequest', lambda e: app.Quit(), when = True)
przycisk.Notify('Pressed', lambda a: app.Quit(), when = False)

# Otwieramy okno i uruchamiamy aplikację
okno.OpenWindow()
aplikacja.Run()

# Niekonieczne Python sam zwolni pamięć
del aplikacja
```

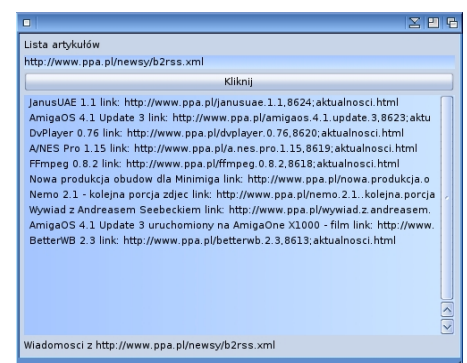


Efektom powyższego kodu powinna być taka prosta aplikacja, jak widzimy po lewej stronie.

Czas na coś trudniejszego i bardziej skomplikowanego.

Nie za duży program

Stworzymy prosty czytnik RSS, pozwalający na odczytanie wiadomości z prawie każdej strony posiadającej tę funkcjonalność. Program będzie składał się z formularza z polem tekstowym, przyciskiem oraz listą `pymui.List()` i



Prosty czytnik RSS



```

#-*- coding: utf-8 -*-
import xml.dom.minidom, urllib, pymui

#zaczynamy od utworzenia grupy, w której umieścimy wszystkie kontrolki MUI
grupa = pymui.Group()

#tworzymy kolejno pole tytułu aplikacji, pole wejściowe, przycisk, listę do przechowywania wiadomości oraz pole informacyjne
tytuł = pymui.Text(Contents = 'Lista artykułów')
url_rss = pymui.String()
przycisk = pymui.SimpleButton('kliknij')
lista = pymui.List()
info = pymui.Text()

#wszystkie nowo utworzone obiekty dodajemy do grupy
grupa.AddChild(tytuł)
grupa.AddChild(url_rss)
grupa.AddChild(przycisk)
grupa.AddChild(lista)
grupa.AddChild(info)

#tworzymy notyfikację dla przycisku, po którego kliknięciu wywołujemy funkcję pobierzRss(), do której przekazujemy adres wpisanej przez nas strony
przycisk.Notify('Pressed', lambda a: pobierzRss(url_rss.Contents), when = True)

#z tekstu wiadomości usuwamy polskie znaki - kodowanie 'UTF-8' w tekście i Unikod Pythona jest nieco skomplikowany i powoduje problemy w połączeniu z PyMUI,
#błąd ten zniknie wraz z dostosowaniem PyMUI do Pythona z serii 3.x
def usunPZnaki(text):
    x = {'\xc3\x93': 'o', '\xc3\x94': 'a', '\xc4\x84': 'A',
          '\xc4\x99': 'e', '\xc4\x98': 'E', '\xc5\x82': 'l', '\xc5\x81': 'L',
          '\xc4\x87': 'c', '\xc4\x86': 'C', '\xc5\x84': 'n', '\xc5\x83': 'N',
          '\xc5\xbc': 'z', '\xc5\xbb': 'Z', '\xc5\xba': 'z', '\xc5\xb9': 'Z',
          '\xc5\x9b': 's', '\xc5\x9a': 'S', '\xc4\x85': '&'}
    for key, value in x.items():
        text = text.replace(key, value)
    return text.encode('utf-8')

#funkcja pobiera stronę z serwera i przekazuje do obiektu parsera xml, sprawdza czy wpisany link istnieje oraz, czy podany url zawiera składnię xml
def pobierzRss(strona_rss):
    try:
        usock = urllib.urlopen(strona_rss.contents)
        dom = xml.dom.minidom.parse(usock)
        info.Contents = 'Wiadomości z ' + strona_rss.contents
        parsujRSS(dom)
    except IOError:
        info.Contents = 'Nie mogę otworzyć Twojego odnośnika.'
    except xml.parsers.expat.ExpatError:
        info.Contents = 'Odnośnik, który podałeś, prawdopodobnie nie jest RSS-em.'

#dwie funkcje pozwalające wyłuskać tekst z xml
def pobierzTextPojedynczo(node):
    parts = [child.data for child in node.childNodes if child.nodeType == node.TEXT_NODE]
    return u"".join(parts)

def pobierzText(nodelist):
    return u"".join(pobierzTextPojedynczo(node) for node in nodelist)

#funkcja parsująca obiekt przekazany w funkcji pobierzRss(), podczas iteracji dodajemy kolejne elementy listy do obiektu pymui.List()
def parsujRSS(dom):
    items = dom.getElementsByTagName("item")
    for i in items:
        title = i.getElementsByTagName("title")
        link = i.getElementsByTagName("link")
        wiadomosc = usunPZnaki(pobierzText(title).encode('utf-8')) + u' link: ' + pobierzText(link).encode('utf-8')
        lista.InsertSingleString(wiadomosc)
    return lista

#tworzymy okno aplikacji o rozmiarach 500px x 400px i dodajemy wcześniej utworzona grupę jako główny obiekt
okno = pymui.Window(Width=500, Height=400, CloseOnReq=True, RootObject=grupa)

#ustawiamy notyfikację dla obiektu, pozwalając po kliknięciu w przycisk zamknąć aplikację
okno.Notify('CloseRequest', lambda *a: aplikacja.Quit(), when = True)

#dodajemy okno i je wyświetlamy
aplikacja = pymui.Application(okno)
okno.OpenWindow()

#wreszcie uruchamiamy gotową aplikację
aplikacja.Run()

```

Teraz wpisz w pole tekstowe adres strony RSS, np. <http://www.ppa.pl/newsy/b2rss.xml>. Jeżeli dostałeś listę artykułów, to znaczy, że się udało. Gratulacje.

Jak uczyć się dalej?

Niestety muszę zasmucić maniaków czytania dokumentacji - autor pakietu PyMUI jeszcze jej nie napisał. Na stronie projektu znajdują się jedynie nieco przestarzałe materiały z poprzednich wersji beta. Ale nic straconego. Wy-

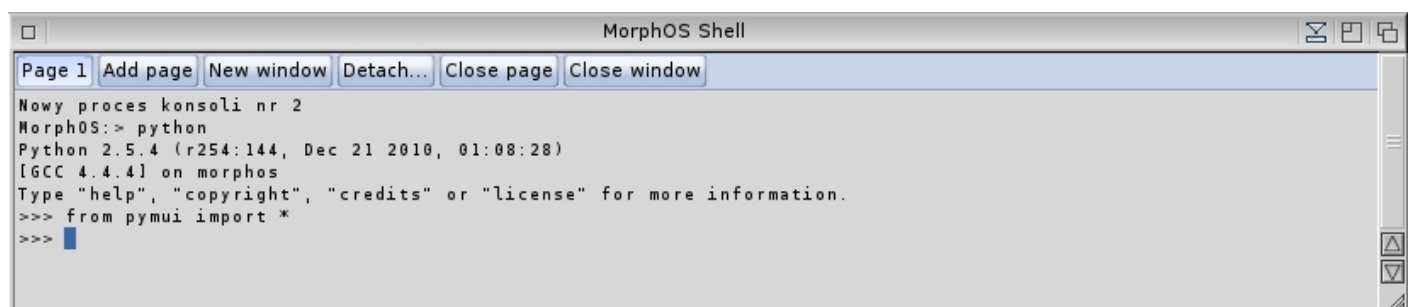
starczy zagłębić się w źródłach pakietu, gdzie znajdziemy uporządkowane funkcje z dość skąpym opisem. Ponadto Yomgui, autor, pisze przy wykorzystaniu PyMUI aplikację Griboullis – program graficzny. Źródła aplikacji możemy pobrać ze strony autora, później ją rozpakować i czytać. Ze swojej strony postaram się, w

miarę wolnego czasu, napisać kolejny artykuł dotyczący PyMUI i tym uzupełnić braki w dokumentacji, a także jeszcze bardziej zachęcić do rozpoczęcia zabawy z Pythonem.

Marcin Spoczyński

Odnosińki

- <http://yellowblue.free.fr/yiki/doku.php/en:dev:pymui:start> - oficjalna strona pakietu PyMUI,
- <http://www.sasg.com/mui/> - oficjalna strona MUI,
- http://library.morphzone.org/Magic_User_Interface_Programming - Tutorial MUI Grzegorza Kraszewskiego,





Amiga i Manga

Manga, jak zapewne wszyscy lub prawie wszyscy wiedzą, to rodzaj bardzo popularnego komiksu, charakteryzujący się oryginalnym stylem rysowania, który zapoczątkowano w Japonii. Styl ten szybko zyskał popularność na całym świecie. Również w Polsce, głównie dzięki serialom anime, takim jak „Kapitan Tsubasa” czy „Dragon Ball”, ludzie zaczęli interesować się japońską kreską. Jak grzyby po deszczu zaczęły pojawiać się w polskich księgarniach co raz to nowsze mangowe komiksy, telewizja raczyła nas codziennymi blokami anime, a w kioskach zaczęła pojawiać prasa dedykowana tej dziedzinie (nieodżałowane „Kawaii”). I właśnie w „Kompendium Kawaii” natknąłem się na ciekawy materiał dotyczący mangi i naszej kochanej Amigi, który został napisany przez kolegę

znanego chyba wszystkim użytkownikom PPA (przy okazji - pozdrawiam). Jako że od tego czasu minęło dobrych parę lat, a temat nie pojawił się w żadnej amigowej prasie, postanowiłem lekko go odświeżyć, nawiązując do tamtego świetnego artykułu.

Zaraz ktoś zapyta, a co z tym wspólnego ma nasza przyjaciółka? Otóż już wyjaśniam. Jako że wśród fanów mangi byli i są również komputerowi zapaleńcy, nie mogło zabraknąć gier, slideshowów czy animacji nawiązujących do tego charakterystycznego stylu. Również nasza Amiga ma na tym polu wiele do zaoferowania, mimo że są to głównie starsze tytuły, to przez lata nie straciły wiele na grywalności, a dzięki uniwersalnej grafice nawet dziś mają się czego wstydzić. Nawet jeśli już znudzimy się amigowymi tytułami i zapagniemy czegoś świeższego, to otwiera się przed nami świat emulatorów, dzięki którym - głównie na nowszych lub bardziej rozbudowanych Amigach - zapoznamy się z grami, które oficjalnie ominęły naszą platformę. I tak stoi przed nami otworem świat gier z PSX-a, mnóstwo pozycji, które dotychczas były przede wszystkim dostępne na PC. Dzięki M.A.M.E. na naszej Amidzie zagospozdzą wspaniałe gry z automatów, a jeśli ktoś chce poczuć japońskiego ducha i klimat z poprzedniej epoki, mamy stare pozycje z Pegasusa czy Segi oraz świetny emulator MSX-a. MSX to 8-bitowy komputer, który największą furorę zrobił w Japonii w latach 80-tych. W tym artykule chciałbym jednak skupić się wyłącznie na grach dedykowanych naszej przyjaciółce. Mangowe gry na Amidzie obejmują praktycznie wszystkie kategorie, począwszy od gier logicznych, po różnej maści strzelaniny, platformówki czy gry cRPG. Chciałbym zaznaczyć, że wiele gier zostało zrobionych przez europejskie studia i kraju kwitnącej wiśni nawet nie powąchały. Nie przeszkadza to jednak w ogóle w zabawie i zapłaniu mangowego klimatu.

Koniec lania wody, czas na konkrety. Na początek moja ulubiona gierka w stylu anime - mianowicie „Leander”. Na trzech dyskietkach mieści się uroczą platformówkę, w której sterujemy wojownikiem walczącym ze złem. Podobnie mi się klimat tej gry, na uwagę zasługuje



również fabuła. Kolej na „First Samurai” i „Second Samurai”. Ta pierwsza była przełomem wśród platformówek z rewelacyjną, jak na swoje czasy, grafiką i świetną muzyką, a jej największym atutem była płynność animacji i nieliniowa fabuła - pełna zagadek i zwrotów akcji. Również „Risky Woods” jest bardzo fajną grą wydaną przez Electronic Arts w 1992 roku. Zajmuje dwie dyskietki i, o dziwo, do odpalenia jej potrzeba tylko 0,5 MB pamięci. W grze pomagamy gościekowi uwolnić zamienionych w kamienie mnichów. Aby tego dokonać, musimy pokonać wielu przeciwników, którzy zamieniają się w monety, za które możemy kupić różne artefakty. W grze „Sword of Honour” przenosimy się do XV-wiecznej Japonii i wcielamy się w samuraja, który musi zmasać hańbę, jaka spadła na ród jego pana (coś dla fanów „The Last Ninja” i „First Samurai”). Lecąc dalej, mamy serię „Turrican” - tytuł mówi sam za siebie, każdy szanujący się amigowiec w to grał, prawda? Fani „Castlevanii” również mogą pograć w amigową wersję tej gry, jednak jej miodność pozostawia wiele do życzenia. Następną dobrą propozycją to „Switchblade” - dwuczęściowa klasyczna zręcznościówka 2D od chłopaków z Gremlin Graphics.

Murowany hicior to trylogia „Double Dragon”, jak również „Golden Axe” - chyba nie trzeba bliżej przedstawiać tych gier - uliczne mordobicie w najlepszym wydaniu. Kolejne cudo to „Rod-Land” - prześliczna mangowa grafika, troszkę dziecinna, ale komu to przeszkadza. „The Speris Legacy” autorstwa Team 17 to zręcznościówka, w której znajdziemy rozwinięty wątek fabularny z barwną, jak na tamte czasy, grafiką. Dla fanów Zeldy pozycja obowiązkowa. Kolejna gra ze stajni Team 17 - „Apodya”, to tytuł, który swego czasu wywarł na mnie duże wrażenie. Niby zwykła strzelanka - sterujemy o dziwo pszczołą, ale to cudne mangowe interko na początku... „Gunbee F99” to strzelanka wydana w 1999 roku przez niemiecką firmę ACP&TCP. W momencie pojawienia się została niemalosiernie zbesztana przez amigową prasę, dziś spokojnie można ją potraktować jako klasykę z mangową grafiką, czyli to, co tygryski lubią najbardziej. Jeśli o klasykach mowa, to nie można zapomnieć o „R-Type”. Autofire i do przodu, myślenie zos-

tawmy na później. Za to „Bubble Heroes” to gra logiczno-zręcznościowa o cukierkowej grafice, która, trzeba podkreślić, jest naprawdę ładna. Fani RPG mogą pograć w „Realms of Sendai”, która była kiedyś dostępna na jednej z płyt ACS i ma status gry shareware. Słabutka grafika, za to zabawa przednia. Kolejny „erpeg”, tym razem w pełni komercyjny, to „Shadowlands”. Postacie żywcem wyjęte z mangi, sama gra to tradycyjne role playing fantasy z izometrycznym widokiem 2D. Lubicie się pościgać? Proszę bardzo. „Wheelspin” grupy Black Legend to wyścigi samochodowe w stylu „Skidmarks” z mangowym designem w tle. Dla tych, którym jazda się nie znudzi, mogą zapuścić sobie na deser „Chase H.Q.”.

Na Amigę nie mogło zabraknąć oczywiście mangowych mordobic. Seria „Street Fighter” również znana jest wszystkim graczom. Pamiętam wojenki, które toczyli fani „Street Fighter” i „Mortal Kombat” o to, który tytuł jest lepszy. Dla nas, fanów anime, jest to pozycja obowiązkowa. „Elfmunia” również daje radę. Miłośnicy konsoli CD32 także znajdą coś dla siebie. Świetna nawalanka „Shadow Fighter” pozwala poczuć klimat, wzbogacony dodatkowo niezłą muzyką. Kolejna gra znana wszystkim fanom anime to „Akira” - na podstawie kultowego filmu o tym samym tytule. Jeśli się nie mylę, została wydana na wszystkie Amigi, lecz w wersji na CD32 dodatkowo dołączono wstawki animacyjne prosto z filmu. Z nowszych pozycji, dedykowanych głównie Amigom nowej generacji, klimatem nawiązującym do mangi jest „Aquaria” opisywana szerzej w poprzednim numerze naszego pisma czy „Shogo” - gra akcji łącząca cechy symulatora i gier FPP z ogromnymi, jak na tamte czasy, wymaganiami.

Powolutku dochodzimy do końca tego artykułu. Wiem, że przytoczone tytuły to tylko kropla w morzu, mały ułamek tego, co powstało i w co możemy się pobawić. A przecież oprócz gier jest wiele innych gatunków (intra, slideshows, animacje, ziny) nawiązujące tematem do mangowego stylu, które możemy odpalić na naszych Amigach i rozkoszować się nimi do woli. Chciałbym również, by ten tekst wyzwoił wenę twórczą w osobach, które tak jak ja interesują się tym dalekowschodnim stylem, a które mają o wiele większą wiedzę na ten temat (wiem, że na PPA tacy są) i aby go rozwinięli oraz poprawili mnie w kwestiach, które uznają za stosowne. Zaproponujcie tytuły bardziej uniwersalne, gdyż wiem, że skupiłem się na tych najbardziej znanych i popularnych w świecie Amigi. Temat jest bardzo obszerny - a Amiga, jak i Manga na to zasługują.

Łukasz „Vegeta” Jeglorz

P.S. Tekstem, który pchnął mnie do napisania tego artykułu i był dla mnie swoistym przewodnikiem, dzięki któremu zainteresowałem się mangowym graniem na Amidzie, był artykuł napisany przez Misumaru Tenchi. Chciałem Ci podziękować i mam nadzieję, że nie czujesz się rozczarowany.



Amiga Mods Anthology

zbiór prawie idealny

Zbiór modułów na Amigę w moim przypadku zawsze zabierał przynajmniej parę gigabajtów na dysku. No cóż, posiadanie Amigi do czegoś zobowiązuje. Przez jakiś czas po padzie dysku twardego zbiór się znacznie skurczył. Chciałem szybko nadrobić stratę pobierając z sieci paczkę modułów o rozmiarze 29 GB, ale to była lekka przesada. Jeden moduł na sto nadawał się do słuchania. Szybko pozbyłem się tak trudnego w utrzymaniu zbiorowiska dziwacznych tworów. Pewnego dnia na dysk mojego Macintosha wpadł zestaw „Amiga Mods Anthology”, który jest zbiorem modułów zebranych przez grupę zapaleńców z TOSEC. Zbiór został udostępniony

na 4 obrazach płyt CD, podzielonych alfabetycznie na wykonawców, a ostatni, czwarty krążek zawiera luźny zbiór modułów nagradzanych na imprezach, najpopularniejszych grup scenowych, utworów w klimacie chiptune, trochę staroci oraz zestaw muzyki w mniej popularnych formatach m. in. PlaySID, Octamed, DigiBooster.

Co tam znajdziemy? Sam listing modułów zająłby wszystkie strony tego magazynu. Na tych płytach jest prawie cała muzyczna historia Amigi. Znajdziemy na niej naprawdę wspaniały zestaw historycznych już modułów, takich sław jak 4mat, Walkman, 911, Nightshade, Audio-monster, Gryzor i tysiące innych. Oczywiście nie zabrakło miejsca dla Polaków. Mamy Scor-



pika i paru innych muzykantów. Wśród nich niestety nie znalazłem JazzCata, który jest jednym z moich faworytów, no ale widocznie twórcy nie chcieli go umieścić albo zwyczajnie przeoczyli. W niektórych katalogach autorów znajdziemy dodatkowy katalog Docs, gdzie umieszczone są informacje, biografia, zdjęcia (nawet te rodzinne), votki z imprez i inne ciekawe dokumenty. Wszystkie katalogi z modułami zawierają listing w postaci pliku txt, gdzie znajdziemy opis katalogu, wielkość pliku oraz komentarz do utworu.

Dodatkowym atutem zbioru jest zawartość katalogu Tools, gdzie znajdziemy oprogramowanie do odtwarzania plików znalezionych na tym zestawie płyt w wersjach na bardziej i mniej

popularne systemy operacyjne. Zestaw zawiera także aplikację mafind dla Amiga-OS, gdzie po wpisaniu nazwy modułu, pliku czy autora aplikacja wskaże nam katalog, w którym mamy go szukać. Część oprogramowania jest już ze względu na okres wydania zbioru nieco nieaktualna, ale z tej sytuacji możemy szybko wybrać szukając w internecie nowszej wersji programu niż ta załączona.

Czy warto ten zestaw pobrać? Myślę, że każdy amigowiec powinien posiadać ten zbiór na płytach i jak najczęściej do niego wracać. Na dyskach jest umieszczone i skatalogowane paręnaście lat amigowej historii, która ciągle trwa, gdyż trac-

ciągłe się bronią przed coraz to bardziej rozbudowanymi edytorami muzycznymi, a obecnie najpopularniejszy Tracker - Renoise - dostępny na Macintosha i PC, cieszy się dość dużą popularnością wśród wielu producentów. Co do amiświata, niestety jesteśmy jak zwykle trochę opóźnieni, ciągle czekamy na DigiBooster 3, dzięki któremu świat amigowych modułów pewnie trochę odżyje. A jak nie, to zawsze pozostanie nam do odpalenia stary i wysłużony Protracker.

Marcin Spoczyński

Konkurs na krzyżówkę

Przypominamy, że konkurs na sporządzenie krzyżówki, zapoczątkowany w pierwszym numerze naszego magazynu trwa nadal. Oczekujemy, że będzie to praca sensowna, związana tematycznie z profilem czasopisma oraz portalu. Rodzaj i rozmiar krzyżówki dowolny (liczymy na Waszą inwencję, akceptujemy standardowe krzyżówki, jolki, panoramiczne, rebusy, anagramy). Prace należy nadsyłać na adres redakcji w postaci plików graficznych (format PNG) wraz z określeniami oraz hasłami (odpowiedziami) w osobnym pliku tekstowym (format ASCII). Prosimy nie uzupełniać plików graficznych hasłami (odpowiedziami)!

Najlepsze prace zostaną nagrodzone publikacją krzyżówki na łamach magazynu, a wytypowany przez jury redakcyjne zwycięzca otrzyma dodatkowo nagrodę.

Redakcja zastrzega sobie prawo do zmiany objaśnień w krzyżówce przed publikacją, nieopublikowania krzyżówek, jeżeli uzna to za zasadne lub, w przypadku niskiego poziomu nadesłanych prac, całkowite anulowanie konkursu i nie przyznanie nagrody.

Adres redakcji: redakcja@ppa.pl

Rozwiązanie konkursów

W odpowiedzi na nasz konkurs wpłynęły artykuły, które pozwoliły zapisać numer, który właśnie trzymacie w rękach. Podobnie jak i poprzednim razem, spośród opublikowanych w numerze artykułów wybraliśmy ten, który naszym zdaniem zasłużył na specjalne wyróżnienie. Wybór zwycięzcy nie był łatwy, lecz ostatecznie, w drodze głosowania, pierwsze miejsce przyznaliśmy **Stanisławowi Sędkowskiemu** za artykuł „*Odprowadzamy czary, czyli krótki poradnik jak ustrzec się od potencjalnej śmierci Amigi*”. Autor został doceniony za oryginalne, zróżnicowane i ciekawe podejście do tematu. Gratulujemy. Z autorem skontaktujemy się w celu ustalenia nagrody. Pozostałym uczestnikom dziękujemy za nadesłane artykuły.

Chwileczkę, wydarzyło się stanowczo zbyt wiele... Harkonnenowie ratują mnie przed porwaniem - nie bezinteresownie co prawda - ale próbowali. Policja, trybik w maszynie Starożytnych, o nie mało dobrała mi się do skóry... Czyżby więc Starożytnym zależało tak bardzo na spotkaniu ze mną? Czemu Los mi nie sprzyja?

Ostatnio jestem trochę na bieżąco jeżeli chodzi o waszą politykę. Ot, pojawił się w kablówce piracki kanał informacyjny z waszego wymiaru. Z tego co mi wiadomo, Los został oskarżony o naruszenie zasad niezawisłości. Przyjmował łapówki i zlecał dzięki ich „mocy sprawczej” wyroki niezgodne z jego poglądami, sumieniem i etyką zawodu. Ukrywa się obecnie gdzieś w galaktyce...

No tak... Cała ta historia na kilometr śmierdzi tanim filmem klasy B! Los otrzymuje zlecenie. W roli najemników zatrudnia Starożytnych, umywając przy tym ręce. Tylko od kogo te zlecenie? Wrogów wciąż przybywa, jak to w showbiznesie... Rozsądek mówi mi, że najlepsze co mogę teraz zrobić to tu zostać!

...i tym samym podpisać na siebie wyrok? Proszę nie zapominać co to za miejsce. Otwarcie tunelu ostatni raz nastąpiło dwa lata temu. System L·I·N·C poinformował, iż kolejne otwarcie nastąpi za parę dni. Ale mam dla Pani propozycję. Tutejsi emerytowani bohaterowie nie obrazili by się gdybym pożyczył kilka przydatnych gadżetów. Mamy broń prosto z filmu Imperial z Nowej Europy, znajdzie się też kuloodporny kombinezon... Myślę, że pomogłyby to Pani wyjaśnić kilka osobistych spraw.

Przekonał mnie Pan!

Rysunki i scenariusz: Bartek Żołyński (BagoZonde)

Czy niezawodny japoński system tuneli zda egzamin? Gdzie trafi uzbrojona po zęby bohaterka? Czy trafi na trop prowadzący do jej prześladowcy? Wszystko zależy od Ciebie, oddaj głos w ankiecie na portalu PPA.