



Polskie Pismo Amigowe

Numer 4/2014 (14)

cena 20 zł



ISSN 2354-0273



9 772354 027002

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, co pozwoli wypełnić dwie strony pisma.

Redakcja zastrzega sobie prawo do wydrukowania każdego ze zgłoszonych artykułów w „Polskim Piśmie Amigowym”. Artykuł wybrany do druku nie może być nigdzie opublikowany ani przed, ani rok po wydaniu go drukiem w PPA.

W miarę możliwości będziemy starać się opublikowane artykuły nagradzać albo w drodze plebiscytu na najlepszy tekst numeru lub w postaci wierszówek wypłacanych za zapełnioną stronę pisma.

Oprócz tego, tradycyjnie, każdy autor artykułu zakwalifikowanego do druku otrzyma bezpłatny egzemplarz pisma.

Termin nadsyłania artykułów upływa:

31 stycznia 2015 roku.

Zapraszamy do udziału!



Polskie Pismo Amigowe

Redaktor naczelny: Sebastian Rosa

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

Współpracują: Mateusz Eckert,
Aleksander Giedyk, Tomasz Pacyna

Skład: Sebastian Rosa

Grafika: Sławomir Woźniak

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

Skład pisma wykonywany jest w programie OpenOffice 3.

Druk: Drukarnia cyfrowa DCTF w Tarnowie.

Wydawnictwo AIBB – Tomasz Bernacik
ul. Piotrkowska 201 lok.10, 90-451 Łódź



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 2010–2014.
Tux (maskotka Linuksa) © Larry Ewing,
Simon Budig i Anja Gerwinski.

<http://www.ppa.pl>

W numerze

	WHDLoad	4
	Gry – nowości	8
	FCEU	11
	Rozmowa z Tomaszem Golikiem – właścicielem firmy TimSoft ..	12
	Synergy	13
	Tales of Gorruth	14
	Nasza pierwsza gra – kurs programowania pod AmigaOS – część 9	16
	Cave Story	18
	Full Motion Video dla CD32	20
	A604n – rozszerzenie pamięci dla A600	21
	RetroKomp 2014	22
	Riverwash 2014 Party Raport	24

Od redakcji

Trzymasz właśnie w rękach czternasty numer naszego periodyku, który pozwolił zachować trzymiesięczny cykl wydawniczy (numer udało się złożyć w grudniu 2014 roku, ale do Ciebie trafił zapewne dopiero na początku stycznia 2015 roku). W numerze znaleźć można kilka ciekawych, a co najważniejsze świeżych lub będących wiecznie na czasie tematów.

Na początek polecam obszerny artykuł dotyczący WHDLoad. Wciąż jest wiele pytań dotyczących tego programu, wiele osób, zwłaszcza początkujących, ma nadal z nim problem. Mam nadzieję, że ten artykuł trochę pomoże wyjaśnić kilka wątpliwych kwestii, zwłaszcza, że całkiem niedawno miała miejsce premiera nowej wersji.

Idąc dalej, rozpoczęliśmy nowy cykl przeglądu nowych gier, których całkiem sporo pojawiło się w 2014 roku. Nie wszystkie zasługują na recenzję, ale na wzmiankę w tego typu rubryce jak najbardziej. Częstotliwość pojawiania się tego typu kolumny uzależniona będzie oczywiście od tego, co będzie się pojawiać. Ponadto nie mogło zabraknąć obszerniejszych recenzji gier, których co najmniej dwie pojawiają się w każdym numerze. Tym razem przeczytać możecie o „Cave Story”, a także o rodzącej wiele pozytywnych, jak i negatywnych opinii grze „Tales of Gorruth”.

Waszą uwagę proponuję również zwrócić na dwa raporty z amigowo-scenowych imprez: RetroKomp 2014 oraz Riverwash 2014. Dużo ciekawego na nich się działo, a autorzy tekstów starali się to wszystko ładnie ubrać w słowa i zaprezentować tym, którzy nie mogli się stawić. Pozostając w temacie publicystyki, jeden z naszych czytelników dotarł do Tomasa Golika z TimSoft i przeprowadził z nim wywiad, który również publikujemy.

Nie zapomnieliśmy także zarówno o posiadaczach Amig klasycznych oraz użytkownikach systemu MorphOS. Dla tych pierwszych mamy recenzję najnowszego rozszerzenia pamięci do Amigi 600 – A604n oraz tekst poświęcony rozszerzeniu Full Motion Video dla Amigi CD32. Z kolei osoby spod znaku niebieskiego motyla na pewno zainteresują się rzutem oka na emulator konsoli Pegasus – FCEU oraz jakże przydatny program-pchelkę – Synergy.

W numerze publikujemy również ostatnią część kursu programowania pod AmigaOS. Nie oznacza to jednak, że kończymy tę rubrykę. Od następnego numeru rozpoczynamy nową szkodkę dla programistów. Szczegóły – za kwartał.

Kończąc te kilka słów wstępu, w imieniu swoim i całej redakcji, życzę wszystkim naszym czytelnikom wszelkiej pomyślności w Nowym Roku i mam nadzieję, że będzie nam dane spotkać się na łamach kolejnych czterech numerów.

Sebastian Rosa

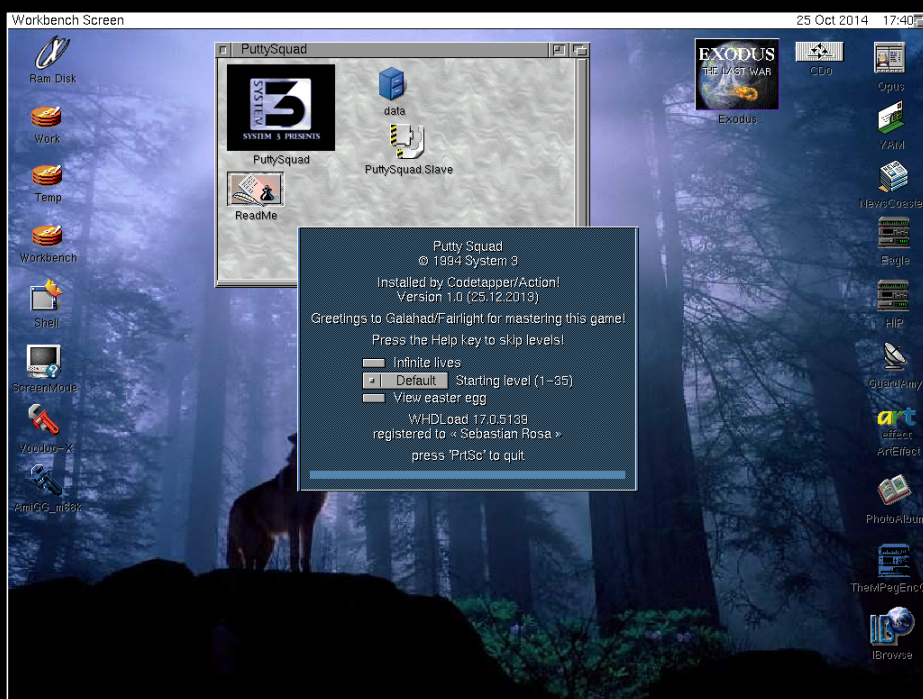




WHDLoad

Amiga początkowo utożsamiana była z grami, które rozprowadzane były na dyskietkach. Z czasem ten nośnik przestał być funkcjonalny, zarówno ze względu na prędkość odczytu danych, pojemność, jak i żywotność. Dodatkowo rozszerzenia sprzętowe, w które amigowcy wyposażali swoje „przyjaciółki” zaczynały ograniczać działanie starszego oprogramowania, a wachlowanie dyskietkami w dobie twardych dysków przestało być wygodne. Naturalną kolejną rzeczą było więc pojawienie się czegoś, co ułatwi trudny żywot tym, którzy lubią czasami w coś zagrać. Zaczęły więc pojawiać się tzw. HD łatki, które umożliwiały „oszukiwanie” gry, aby nie odwoływała się na sztywno do urządzenia DfX: w celu odczytania danych, pozwalając tym samym na uruchomienie gry z twardego dysku. To rozwiązanie było zadowalające, lecz bardzo szybko okazało się, że posiada pewne wady. W większości przypadków taka łatka działała z jedną, konkretną wersją gry – głównie piracką. Osoby, które posiadały wersję oryginalną, dystrybuowaną na zabezpieczonych przed kopiowaniem dyskietkach, nadal miały problem, gdyż takiej gry nijak nie szło przenieść na dysk twardy, nawet pomimo dołączonych do „łatki” narzędzi tworzących obraz dysku. Dodatkowo bardzo często taka łatka działała w obrębie pewnych konfiguracji sprzętowych. Zaistniała więc w ten sposób potrzeba narzędzia, które nie tylko „oszuca” grę, aby czytała dane z innego miejsca, ale również aby gra „myślała”, że jest uruchamiana na domyślnie ustalonej konfiguracji, bez znaczenia jaka ona w rzeczywistości jest. W ten właśnie sposób narodziły się HDLoadery z funkcją degradera, które pozwalają odłączyć lub wyłączyć niektóre komponenty systemu, aby możliwe było uruchomienie danej aplikacji w tak jakby emulowanym środowisku innej maszyny. Jak wielu widać, jest to kolejne „oszustwo”, którym gra jest traktowana. Amigowcy software market opanowały dwa takie programy: JST oraz WHDLoad. Ich cel był dokładnie taki sam, lecz filozofia działania, choć różniła się, okazała się lepsza i bardziej przyszłościowa wyłącznie u jednego z nich – WHDLoad. Autor JST szybko to dostrzegł, porzucił jego rozwój i dołączył do ekipy wspierającej rozwój bohatera niniejszego artykułu.

Odwagę się stwierdzić, że WHDLoad to statystycznie najpopularniejszy i najczęściej uruchamiany amigowy program. I choćby znalazło się kilka, kilkanaście lub kilkadziesiąt osób, które być może z niego nie korzystają, to na pewno stanowią one niewielki odsetek wszystkich amigowców, którzy o WHDLoad musieli chociażby słyszeć. Autorem programu jest niemiecki programista Bert Jahn, który pierwszą, publiczną wersję beta programu wydał w sierpniu 1996 roku, podczas gdy pierwszą wersję 1.1 prawie dwa lata później, w maju 1998, jednocześnie szybko zmieniając sposób numerowania wersji, co sprawiło, że w lipcu 1998 otrzymaliśmy wersję 7.0. Początkowo jego działanie było bardzo skromne i ograni-



W momencie uruchomienia gry/dema/programu WHDLoad wyświetla tzw. splash screen, gdzie możemy zobaczyć między innymi nazwę uruchomionego programu, jego autora, a od wersji 17.0 także możliwość aktywacji specjalnych funkcji slave oraz pasek postępu wczytywania danych.

czało się wyłącznie do jego podstawowej funkcjonalności, aby zapewnić możliwość uruchamiania gier z twardego dysku. Z wersji na wersję jego możliwości jednak przybywało, a program rozwijany jest aż do dziś.

Co to jest?

WHDLoad to, jak już wspomniałem, program, który zwykło nazywać się mianem degraderów, czyli narzędzi służących do „odłączania” niektórych komponentów „silniejszych” konfiguracji w celu uruchomienia starszych gier lub programów (to tak pobieźne, dla laików). WHDLoad posiada jednak znacznie większe możliwości, gdyż działa niczym emulator, który uruchamia zazwyczaj gry nie z dysku, lecz ze specjalnie przygotowanych obrazów dysków. Dodatkowo WHDLoad potrafi „uaktywnić” nieistniejące w oryginalnej grze elementy, takie jak na przykład zapisywanie tabeli najlepszych wyników, zapewnić wyjście z gry z powrotem do systemu czy uaktywnieniu tzw. cheatów. Idea działania zainstalowanych gier lub programów przy pomocy WHDLoad oparta jest na zasadzie „pana i niewolnika”. Niewolnikiem jest plik **slave**, który zawiera szereg instrukcji i poleceń interpretowanych przez WHDLoad (pana), pozwalających ominąć lub podmienić wiele elementów, które sprawiają problemy na innych konfiguracjach niż pierwotnie zakładali twórcy danej gry.

Jak korzystać z WHDLoad?

Na początek należy upewnić się, że nasza konfiguracja sprzętowa posiada niezbędne do działania WHDLoad wymagania. Program nie jest wcale wybredny i zadowolony się każdym procesorem linii 68k (choć pewne funkcje są dostępne dopiero dla 68010), kickstartem 2.0 oraz minimum 1 MB pamięci. Gdy to mamy już za sobą, czas pobrać archiwum. Do wyboru mamy dwie opcje: archiwum użytkownika (USR) lub dewelopera (DEV). Proponuję zainteresować się tym pierwszym i korzystając ze znajdującego się tam skryptu instalacyjnego, zainstalować program. W zasadzie wszystko co niezbędne to raptem kilka plików (**WHDLOAD**, **DIC**, **RawDIC**), które zostaną umieszczone w katalogu C: znajdującego się w archiwum WHDLoad do katalogu C: znajdującego się na partycji systemowej (wskazywanego domyślnie przez C:) lub innego, wskazanego przez polecenie PATH. Posiadanie WHDLoad na twardego dysku to jednak nie koniec.

Niektóre pliki **slave** używają tak zwanego trybu **kickemu** w celu zapewnienia kompletnego

(czytaj – bardziej wiernego) środowiska dla zainstalowanego programu. Do tego celu tryb kickemu wymaga pliku z obrazem oryginalnego Kickstartu, który należy rzucić (na przykład programem *GrabKick*) z naszej Amigi. Obrazy Kickstartu 1.2 i 1.3 mogą być zgrane tylko z komputera Amiga 500, podczas gdy wersja 3.1 może być zgrana z komputerów Amiga 500/600/1200 oraz Amiga 2000/3000/4000, które są wyposażone w kości ROM z Kickstartem 3.1. Do kompletności brakuje nam jeszcze plików relokacji (RTB), które można znaleźć w aminetowym archiwum *SKick*. Tutaj pierwsza uwaga, gdyż jest to jeden z najczęściej popełnianych błędów przy instalacji WHDLoad. Zrzucony obraz Kickstartu musi zostać odpowiednio nazwany, a nazwa ta jest podana w dokumentacji WHDLoad i jest uzależniona od wersji i pochodzenia naszego Kickstartu. Druga uwaga jest taka, że pliki relokacji muszą odpowiadać naszej wersji Kickstartu i muszą również być odpowiednio nazwane. Zarówno plik/pliki obrazu Kickstartu oraz odpowiadające im pliki relokacji należy umieścić na partycji systemowej w katalogu *DEVS:Kickstarts*.

Na koniec, gdy już zdecydujemy się na zarejestrowanie programu, pozostaje umieszczenie w katalogu C:, L: lub S: klucza rejestracyjnego.

Instalujemy grę

Kolejnym krokiem jest zainstalowanie gry, dema lub programu, który chcemy uruchomić. Ze strony programu należy pobrać archiwum z interesującym nas pakietem instalacyjnym. W nim znajduje się kilka plików, które mogą być inne dla każdej z gier, którą będziemy instalować, lecz trzy z nich są obowiązkowe:

- plik *slave*, kluczowy element pakietu indywidualnie napisany dla każdej obsługiwanej gry,
- plik *readme*, w którym zapisane są m. in. informacje o wymaganiach gry oraz specjalnych ustawieniach i funkcjach *slave*,
- skrypt instalacyjny wraz z ikonką, przy pomocy którego należy zainstalować grę/demo/program na twardym dysku.

Opcjonalnie w archiwum może znajdować się także jeszcze jeden plik – z rozszerzeniem



slave, który służy w większości przypadków do zainstalowania gier zabezpieczonych przed kopiowaniem, a także dodatkowe ikonki w różnych standardach, dokumentacja do gry, sposób przejścia, porady i inne.

Po rozpakowaniu archiwum należy zapoznać się z wymaganiami *slave*. Jest to bardzo istotny element każdego *slave*, o czym wielu użytkowników WHDLoad zapomina. W przypadku WHDLoad nie ma znaczenia, że gra oryginalnie śmiga na A500 z 512 kB Ram. Budowa i metoda działania programu sprawia, że taka gra może bezwzględnie wymagać np. 1 MB pamięci Fast lub 0,5 MB pamięci Chip i 0,5 MB pamięci Fast (co niejako stanowi minimalne wymagania samego programu). Inna może dać się uruchomić na nierozbudowanej A600, lecz z wyłączoną funkcją *PRELOAD* (o tym później) i aby ją włączyć potrzebujemy dodatkowej pamięci. Przeczytanie tych kilku linii tekstu czasami potrafi zaoszczędzić nam wiele czasu na analizowanie powodów niedziałania gry, a dodatkowo możemy dowiedzieć się, na przykład, o nowych funkcjach wbudowanych w

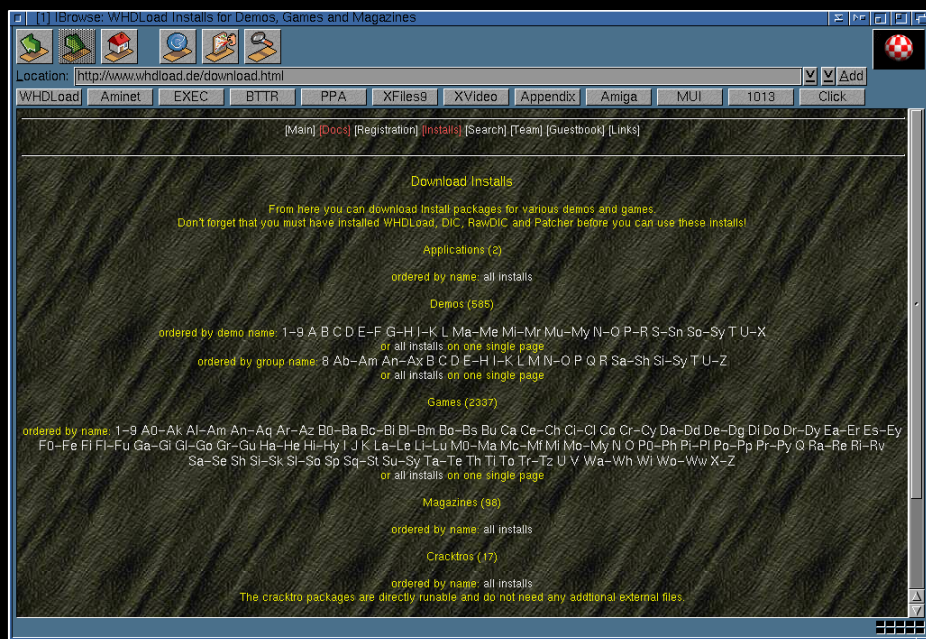
slave (np. zapisywaniu tablicy najlepszych wyników, co w oryginalnej grze uruchamianej z dyskietek nie miało miejsca).

Gdy już wiemy, że nasz system sprosta wymaganiom instalowanej gry, możemy rozpocząć proces instalacji gry. Uruchamiamy skrypt instalacyjny i postępujemy zgodnie z wyświetlanymi tam poleceniami. Między innymi wskazujemy źródłowe urządzenie, z którego dane będą kopiowane (do wyboru możemy mieć np. fizyczną stację dyskietek (DF0, DFx, RAD czy też plik ADF), miejsce docelowe na dysku twardym, gdzie zostanie zainstalowana gra oraz wybieramy stosowną ikonkę (jeżeli są jakieś do wyboru). Niektóre skrypty pozwalają również ustawić odpowiednie parametry instalowanej gry, które zapisują się w tooltypach ikony. Warto mieć również na uwadze, że bez posiadania oryginalnych dyskietek z grą lub oryginalnych obrazów dyskietek (np. w postaci plików IPF), w większości przypadków nie będziemy w stanie zainstalować gry. Twórcy *slave'ów* do WHDLoad wyszli z założenia, że nie ma sensu wspierać pięciu scrackowanych wersji gry i lepiej skupić się na jednej, która jest dokładnie w takiej postaci binarnej, w jakiej napisali ją jej autorzy. Dodatkowo, z początkiem istnienia WHDLoad pod wątpliwość stawiano jego bycie w zgodzie z prawem autorskim i dyskutowano czy program go przypadkiem nie łamie. Na szczęście dla użytkowników i autora tego typu zarzuty okazały się bezpodstawne.

Po zakończeniu procesu instalacji możemy spróbować uruchomić grę z katalogu, do którego nakazaliśmy jej instalację. W przypadku problemów z uruchomieniem należy upewnić się, że nasza konfiguracja spełnia wymagania stawiane przez *slave* lub czy nie są wymagane specjalne ustawienia, o których wspomniane jest w pliku *README*. A co jeżeli sprostaliśmy wymaganiom, a gra nadal nie chce działać? Pozostaje...

Zabawa parametrami

Z uwagi na swoją mnogą funkcjonalność WHDLoad zapewnia ogromne możliwości konfiguracyjne, zarówno pod kątem działania sterowanej przez *slave* gry, jak również dodatkowych funkcji. Te pierwsze czasami trzeba stosować w różnych kombinacjach i są one różne



Na stronie WHDLoad znajdziemy pokaźny katalog tytułów, do który przygotowano stosowny plik *slave* z pakietem instalacyjnym. Wszystko jest posegregowane alfabetycznie.



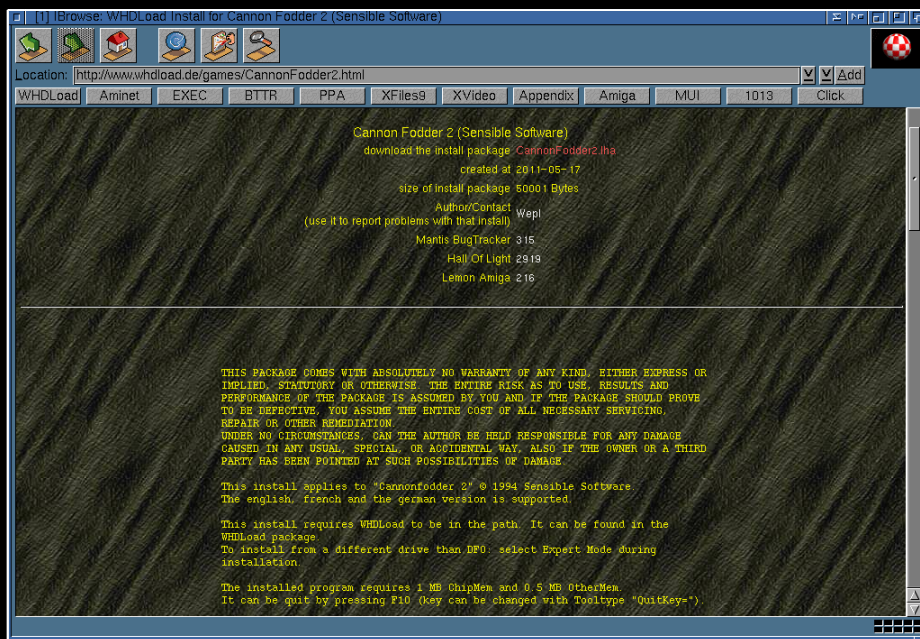
dla różnych konfiguracji sprzętowych i procesorów, w które są wyposażone. Parametry, w zależności od sposobu uruchamiania gry, dema lub programu mogą znajdować się w to-oltypach ikony lub być podawane jako parametr komendy CLI. Część z tych parametrów możemy również potraktować jako globalne, czyli znajdujące zastosowanie dla każdego slave (uruchamianej gry) bez potrzeby ich wpisywania w ikonke lub jak parametr przy uruchamianiu z CLI. Wówczas należy wpisać odpowiedni parametr (lub zmodyfikować istniejący) w pliku **WHDLoad.prefs** znajdującym się w katalogu S: partycji systemowej.

Lista parametrów jest bardzo długa i dobrze opisana w dokumentacji programu. Najpopularniejsze, z którymi zetknie się praktycznie każdy użytkownik to:

- **PRELOAD** – funkcja do wczytania całej gry do pamięci, dzięki czemu doczytywanie danych w trakcie gry jest praktycznie niezauważalne dla użytkownika. Brak aktywacji tej funkcji pozwala na uruchomienie większości gier na słabszych konfiguracjach, lecz należy liczyć się z tym, że WHDLoad doczytując dane będzie zmuszony połączyć się z systemem, aby wykonać operację dyskową, a to skutkować będzie charakterystycznym migotaniem ekranu;
- **SLAVE=nazwaslave.slave** – ten parametr jest obowiązkowy dla każdego slave i wskazuje nazwę pliku slave, który ma zostać wywołany przez WHDLoad;
- **QUITKEY=\$xx** – każdy slave posiada domyślnie ustawiony klawisz, który powoduje wyjście do Workbench'a z uruchomionej gry lub dema. Ten parametr pozwala ustawić inny klawisz wyjścia (funkcja bardzo przydatna dla posiadaczy A600, którzy nie posiadają klawiatury numerycznej, a bardzo często domyślnie klawisze wyjścia przypisane są do tam znajdujących się klawiszy);
- **NOCACHE** – wyłączenie cache procesora. Funkcja zazwyczaj przydaje się, gdy gra działa zbyt szybko lub występują artefakty graficzne na ekranie. Posiadacze procesorów 68040 i 68060 mogą również korzystać z dodatkowych funkcji związanych z cache procesora, takich jak **CHIPNOCACHE**, **BRANCHCACHE**, **DCACHE**;
- **CUSTOM** – parametr służący do kontroli specyficznych elementów slave. Dla przykładu autor slave mógł zapisać w nim, że zastosowanie **CUSTOM=1** będzie skutkowało włączeniem nieskończonej liczby żyć, a **CUSTOM=2** brak kolizji. W innych tytułach mogą one oznaczać rozpoczęcie gry od podanego etapu lub włączenie jakiejś ukrytej funkcji.

WHDLoad i CD32

Ukłonem autora w kierunku posiadaczy Amigi CD32 jest specjalna wersja WHDLoad przygotowana właśnie dla tej maszyny. Została zoptymalizowana pod ten sprzęt i uruchomi się jedynie na „gołej”, nierozbudowanej Amidze CD32. Pakiet nie uruchomi się na tym komputerze z dodatkiem SX-32, jak i na żadnej innej maszynie. WHDLoadCD32 nie wymaga rejestracji, ale jednocześnie oferuje znacznie mniej w stosunku do oryginału. Nie posiada wielu funkcji uruchamiających i emulujących pewne środowiska, nie zapisuje wyników ani nie adresuje więcej niż 2 MB pamięci Chip. Wydaje się być bardzo okrojona wersją, jaką niektórzy mogliby określić mianem dema lub bety. Pomimo, że można odnieść takie wrażenie



Należy uważnie i ze zrozumieniem czytać dokumentację do każdego pliku slave. Znajdują się w nim informacje dotyczące wymagań, które musi spełniać nasza konfiguracja sprzętowa, aby możliwe było uruchomienie gry/dema/programu. Wymagania plików slave nie są adekwatne do wymagań samej gry!

nie, to jednak tak nie jest, a WHDLoadCD32 to odpowiednie dostosowanie programu do działania w określonym środowisku – w tym przypadku na Amidzie CD32.

Co jeszcze warto wiedzieć?

WHDLoad to oprócz głównego programu i plików slave do obsługi poszczególnych tytułów, również szereg innych narzędzi, z którymi spotkamy się albo podczas instalacji, albo gdy nieco bardziej zaczniemy zgłębiać jego tajniki. Już wspominałem o czymś takim, jak **DIC** i **RawDIC**, ale może warto napisać o nich kilka słów. Mianowicie, WHDLoad sam w sobie uruchamia grę lub demo, ale najpierw należy ją przeciwieście zainstalować. W przypadku tzw. gier „dosowych” sprawa wygląda prosto, gdyż skrypt instalacyjny po prostu kopiuje pliki z dyskiety lub ich obrazów (choć nie zawsze) we wskazane miejsce do specjalnie stworzonego tam katalogu **DATA**. Jeżeli mamy do czynienia z dyskami zapisanymi w formacie NDOS, sprawa już wygląda nieco inaczej i właśnie w tym momencie do akcji wkraczają narzędzia pomocnicze. Najpopularniejszym jest **DIC**, który wykonuje zrzut fizycznego dysku znajdującego się w napędzie DFx: do postaci obrazu dyskiety o rozmiarze 901120 bajtów. Rozeznani w temacie szybko zauważą, że jest to najzwyczajniejszy plik ADF i zapewne już nie raz kombinowali z podmianą takich plików na ich własne lub tworzyli je na przykład poza skryptem instalacyjnym i po prostu wgrali w odpowiednie miejsce. Czasami zdarza się jednak, że rozmiar takiego obrazu dyskiety jest mniejszy lub większy niż ADF. Co wtedy? Jeżeli mamy do czynienia z pierwszą sytuacją, oznacza to, że autor slave miał do czynienia z grą, która oryginalnie nie zajmowała całego dysku, a na przykład pierwszych 40 ścieżek (a więc połowę). Wówczas nie ma sensu tworzyć pełnego obrazu dysku i poświęcić ponad 400 kB na coś, co będzie wypełnione zerami i przy pomocy programu **DIC** zrzucano wyłącznie pewien obszar. Tego typu rozwiązanie stosowano bardzo często przy pierwszych instalatorach, gdy duże dyski twarde w Amigach nie były jeszcze tak popularne i zanim autor

WHDLoad nie zaimplementował obsługi spakowanych obrazów przy pomocy XPK. Obecnie tego typu sztuczki już raczej nie są stosowane.

A co w sytuacji, gdy rozmiar obrazu jest większy niż standardowego pliku ADF? Tutaj do akcji wkracza drugi wspomniany program, czyli **RawDIC**. Jest to narzędzie służące do tworzenia plików lub obrazów dyskiety z oryginalnych dyskiety zapisanych w ich własnym formacie, w większości przypadków NDOS z funkcjami zabezpieczającymi. Zasadą działania **RawDIC** przypomina WHDLoad, który jest „panem” i potrzebuje do pracy niewolnika – w tym przypadku jest to plik **islave** zawierający wszystkie kluczowe informacje pozwalające „rozkodować” dyskiety. Ale **RawDIC** nie służy tylko do tego. Czasami gra, która oryginalnie była na dyskiecie typu NDOS, może zawierać strukturę plików, którą wykorzystuje WHDLoad i będzie uruchamiać grę nie z obrazów, lecz bezpośrednio z plików znajdujących się w katalogu **DATA** (co pozwala nie tylko zaoszczędzić miejsce, ale również ogranicza wykorzystanie pamięci przez WHDLoad, który nie potrzebuje już alokować tyle pamięci ile zawierają obrazy dyskiety, lecz tyle ile faktycznie zajmują pliki). Z racji tego, że sami we własnym zakresie z dyskiety NDOS nie potrafimy zrzucić plików, **RawDIC** i pomocniczy plik **islave** zrobią to za nas. Przykładem takich gier jest np. „Perihelion” czy też „Putty Squad”, ale tytuły można mnożyć. Warto jednak pamiętać, że tak zrzuconego obrazu ponownie nie „wgramy” na dyskiety (przynajmniej nie przy wykorzystaniu ogólnodostępnych programów, których większość z Was wykorzystuje przy zabawie ADF-ami). Starsze skrypty instalacyjne wykorzystywały również program **Patcher**, który jest uboższą wersją **RawDIC** i nie jest już obecnie wykorzystywany przez nowo pisane slave'y.

Osoby zainteresowane tematem innych narzędzi pomocniczych dla WHDLoad (jak i być może tematem pisania własnych slave) mogą pobrać wspomniane wcześniej archiwum dla programistów (DEV). Znaleźć tam można kilka innych ciekawych programów, jak na przykład program **SP**, który służy do analizy obrazów



zrzutu pamięci WHDLoad w celu wyciągnięcia z niej obrazków czy ITD do przenoszenia obrazów dyskieciek na fizyczne nośniki. Programy tam znajdujące się wymagają już nieco więcej wiedzy technicznej z obszaru WHDLoad, jak i programowania w assemblerze i nie będę się na nich koncentrować.

Jeżeli przyjrzymy się bliżej stronie WHDLoad znajdziemy na niej również sekcję poświęconą wersjom beta programu, które są testowane przez szerokie grono testerów. W tym samym miejscu znaleźć można także starsze wersje programu na wypadek problemów z kompatybilnością (o tym za chwilę), jak również dalsze ciekawostki, takie jak chociażby tzw. **uniwersalny slave** dla programów działających pod systemem. Można zadać pytanie, w jakim celu coś takiego istnieje? Odpowiedź jest bardzo prosta. Po pierwsze, gra lub program mogą działać pod starszą wersją systemu i na obecnie przez nas używaną może wywoływać zawieszanie, a po drugie jej systemowe napisanie sprawia, że nie są wymagane specjalnie „hacki” istniejące w znacznej większości typowych plików slave.

Nie ma róży bez kolców

WHDLoad to program, którego działanie i praca jaką wykonuje są niezwykle skomplikowane i wymagają sporej wiedzy nie tylko programistycznej, ale także i tzw. znajomości Amigi od środka. Dlatego też zasada dotycząca posiadania najnowszej wersji programu na dysku w wielu przypadkach nie zdaje egzaminu. Pliki **slave** są tworzone przeważnie pod mechanizm programu WHDLoad (zazwyczaj najnowsza wersja), jaki ma akurat autor pod ręką w chwili przygotowywania instalatora. W miarę rozwoju programu jego algorytm działania oraz zmiany w API mogą spowodować niekompatybilności z plikami **slave** napisanymi pod stare wersje programu. Objawiają się one np. poprzez pojawianie się okien z błędami, których nie rozumiemy. Warto w takich przypadkach, przy próbie uruchamiania takich pozycji, cofnąć się do poprzedniej wersji WHDLoad. Jak ich wspólnie używać? Można w dwojaki sposób. Pierwszym jest skopiowanie pliku **WHDLOAD** starszej wersji do katalogu z grą, drugim jest umieszczenie pliku **WHDLOAD** pod zmienioną nazwą w katalogu C: (np. **WHDLOAD** w wersji 16.1 możemy nazwać **whdload161**) oraz modyfikacja wpisu w ikonie danej gry i zastąpienie frazy **WHDLOAD** przez tę, która odpowiada wymaganej przez dany tytuł wersji (np. wspomniana wcześniej nazwa **whdload161**). Jak więc widać czasami trzeba się trochę nagimnastykować, ale warto pamiętać o zasadzie, aby jednak problematyczny slave był uruchamiany przez WHDLoad, pod który był stworzony (jeżeli nie ma do slave aktualizacji).

Kolejną niedogodnością może być to, że program wydaje się być nieco skomplikowany w obsłudze. Oczywiście ja bym się tutaj spierał, gdyż osobiście nic w nim skomplikowanego nie widzę, no ale nie każdy musi dobrze czuć się w tych wszystkich parametrach i ustawieniach. Na to jednak wymyślono ułatwienie w postaci tzw. launcherów, czyli odpalaczków, które przeszukują dysk lub wskazany na nim katalog w celu zbudowania bazy dostępnych gier i ich uruchomienie. Do takich launcherów należy zaliczyć **iGame**, **TinyLauncher** czy też **X-bEnCh** (dwa z nich były już opisywane na łamach naszego magazynu). Każdy z nich napisany jest przez inną osobę i charakteryzuje się innymi dla niego elementami, które stanowią jego wyróżnik.

Największy problem, z jakim mogą spotkać się użytkownicy WHDLoad, to konflikt z innymi urządzeniami i oprogramowaniem, które wywołują przerwania w regularnych lub losowych odstępach czasu. Przykładem tutaj są aktywne działające karty sieciowe i urządzenia USB. Rozwiązaniem najprostszym jest wyłączenie stosów TCP/IP oraz USB na czas używania WHDLoad (choć od wersji 16.8 posiadacze urządzeń USB wpiętych w mostki Mediator nie muszą już tego robić). Niestety natura i filozofia działania WHDLoad jest obciążona tą niedogodnością i osobiście nie liczyłbym na rychłe (o ile w ogóle) rozwiązanie tego problemu.

Należy też pamiętać, że większość błędów i problemów, z którymi spotykają się użytkownicy nie wynika z tego, że WHDLoad jest wadliwy, lecz wiąże się z niepoprawną konfiguracją pewnych ustawień systemowych. W dokumentacji jasno zostało napisane, że na potrzeby użytkownika WHDLoad z wykorzystaniem kart CF należy wprowadzić odpowiednie ustawienia parametrów **MaxTransfer** i **Mask** w programie **HDToolBox** dla każdej partycji urządzenia, a użytkownicy pakietu Picasso96 powinni mieć wyłączoną obsługę „FakeNativeModes”. Faktem jest jednak niestety to, że WHDLoad „nie lubi” współpracować z pewnymi rozszerzeniami pamięci i kartami procesorowymi. Z tym problemem autor WHDLoad próbuje walczyć i lista obsługiwanych sprzętu stale się powiększa.

Wnioski

Zapominając jednak o wszelkich niedogodnościach programu (bo ciężko jest je nazwać błędami), trzeba przyznać, że jest to kawał dobrego, amigowego rzemiosła. WHDLoad z całą pewnością zajmuje specjalne miejsce w amigowej historii i nie można zapominać jak duży wkład w amigową społeczność miał Bert Jahn i reszta twórców slave'ów (niektórzy nawet już nie żyją). Warto to docenić i nie żałować tych kilkudziesięciu złotych, aby program zarejestrować (zwłaszcza że czasami można zdobyć u autora całkiem sporą zniżkę). Jest naprawdę wart swoich pieniędzy, zwłaszcza, że większość z Was na pewno będzie programu bardzo często używać... No bo czyż nie

ma większej frajdy niż zagrać rundkę w „Lotusa 2”, uzyskać wpis do listy najlepszych, a później, przy kolejnym uruchomieniu gry próbować ten swój wynik poprawić?

Sebastian Rosa

Rejestracja programu

WHDLoad można zarejestrować bezpośrednio u autora odnajdując na stronie programu bezpośrednią informację o sposobie płatności oraz kwocie. Autor akceptuje wpłaty na rachunek bankowy, jak również płatności za pomocą PayPal. Od czasu do czasu na naszym forum organizowana jest akcja mająca na celu zebranie grupy osób zainteresowanych rejestracją programu. Wówczas autor programu wyraża zgodę na obniżenie ceny pojedynczego klucza rejestracyjnego dla każdej osoby. W ten sposób większość polskich użytkowników posiada zarejestrowany program w cenie nieomalże o połowę niższej.

Pomyślnie zarejestrowanie programu kończy się nadesłaniem pliku-klucza bezpośrednio od autora. Plik należy umieścić w katalogu C:, L: lub S: partycji systemowej, aby cieszyć się w pełni zarejestrowaną wersją programu. Rejestracja programu nie tylko oznacza wsparcie dla autora i podziękowanie za jego pracę – niektóre slave wymagają zarejestrowanej wersji, aby odblokować pewne ukryte funkcje.

Nie opłaca się udostępniać swoich plików-kluczy osobom trzecim, gdyż autor skutecznie tego typu akty piractwa blokuje wraz z każdą nowo wydaną wersją programu. Objawy korzystania z pirackich kluczy to zawieszenia pracy programu (czarny ekran) lub dziwne komunikaty przed uruchomieniem gry.

<http://www.whdload.de>

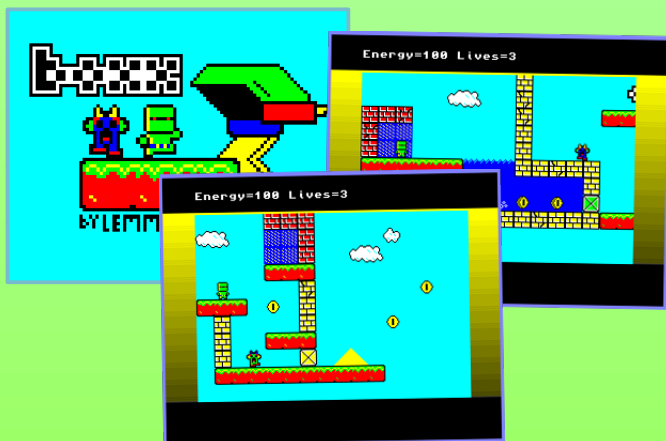




Gry – nowości

Niektóre tytuły wymagają należytych recenzji. Są jednak gry, o których pisanie kilkunastu tekstów nie ma za bardzo sensu – albo z uwagi na mierność pozycji, albo z uwagi na zwyczajne „wyczerpanie” tematu. Tak czy owak, z dziennikarskiego obowiązku należałoby odnotować pewne fakty, zwłaszcza że w ostatnim czasie pojawia się trochę nowego. W tej właśnie rubryce, która mam nadzieję pojawiać się będzie regularnie (choć to zależy od tego, co będzie wychodzić na światło dzienne), będę starać się w kilku zdaniach odnotować ostatnie „grove” wydarzenia i przedstawić moją subiektywną ich ocenę w pięciopunktowej skali. Pamiętajcie jednak, że fakt, iż dana pozycja zostanie tutaj opisana nie oznacza, że poszczególne pozycje nie doczekają się należytej recenzji. Poniższa rubryka przedstawia wyłącznie moją opinię, a Ci którzy się z nią nie zgadzają, mogą to wyrazić sporządzając obszerną recenzję. Zaczynamy.

Sebastian Rosa



Boxx

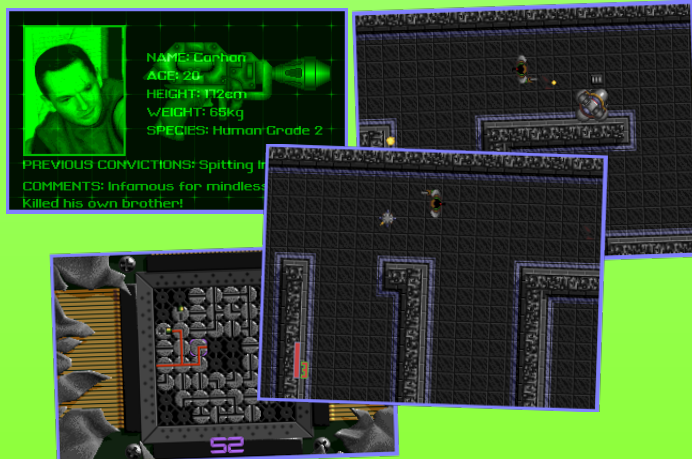
Prosta gra platformowa, utrzymana w graficznym, wypikselowanym, kolorowym stylu retro NES, z kapitalną ścieżką dźwiękową. Celem gracza jest zebranie wszystkich rozrzuconych po pięciu etapach monet i znalezienie wyjścia. Zaprojektowane poziomy są stosunkowo proste, a ukończenie gry nie powinno rodzić żadnych problemów. Największym mankamentem tytułu jest kiepska animacja bohatera oraz powolność działania, co jest oczywiście następstwem tego, że gra powstała w Backbone (a więc AMOS). Jakby jednak nie patrzeć, wśród zalewu produkcji spod tego silnika, „Boxx” wygląda na produkt znacznie bardziej dopracowany i przyjazny użytkownikowi. Gdyby tak tylko było więcej etapów albo na przykład podchwycono silnik z „Mr. Beanbag” lub „Sqrz” mielibyśmy nie lada gratkę. Ocena: 3.

<http://aminet.net/game/jump/Boxx.lha>

Bloodline

To nie jest nowy tytuł, ale rzecz odnaleziona w odmętach twardych dysków. Gra jest strzelaniną, nad którą prace prowadzono w 1993 roku. Stanowi połączenie „Alien Breed”, „Hall of Things” i „Paradroid”. Pomimo tego, że według autora, gra nie jest nawet w stadium alpha, spokojnie można w nią grać (szczerze przyznam, że obecne wersje beta nowych gier wyglądają czasami na bardziej „zacfane”). Biegamy po pokładzie stacji kosmicznej niszcząc krążące po niej roboty. Dostępna jest również ukryta mini gra, która miała stanowić małe oderwanie od ciągłego strzelania, a i na pewno miała mieć jakiś swój ukryty fabularny sens. Pomimo, że gra podobna do wielu później i wcześniej wydanych, jest ciekawym kąskiem – zwłaszcza, aby porównać jakość kodu obecnych i tamtejszych produkcji. Szkoda, że nie została dokończona. Z uwagi, że to nie jest pełnoprawny tytuł, pozostawiam bez oceny.

<http://www.gamethatwerent.com/2014/09/bloodline-amiga-game-recovered-and-released/>



Where Time Stood Still

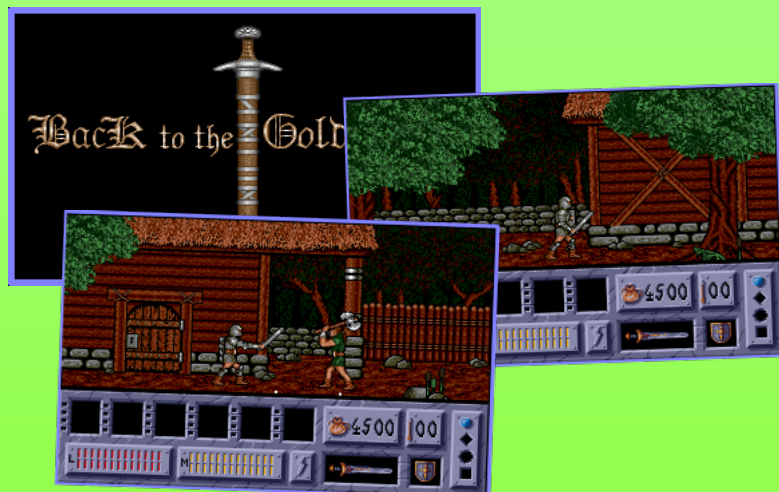
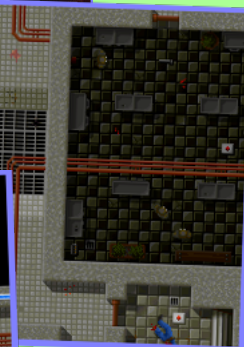
Kolejna konwersja tytułu z Atari ST, którą przygotował Galahad na zasadzie inżynierii wstecznej. Tytuł oryginalnie został wydany przez firmę Ocean. Grę można zaliczyć do kategorii gier przygodowo-zręcznościowych. Zadaniem gracza jest zapewnienie przetrwania rozbitkom katastrofy lotniczej, którzy znajdują się w miejscu dalekim od im znanej cywilizacji. Środowisko gry zbudowane zostało na bazie świata zaprezentowanego z widoku izometrycznego. Osobiście produkcja nie do końca przypadła mi do gustu – trochę posępna, czarno-biała graficznie, z nieco irytującym sterowaniem. Gra powinna uruchomić się na każdej Amidze wyposażonej w 1 MB pamięci. Ocena: 3 (za pełny produkt, który wygląda bardzo profesjonalnie).



Chaos Guns

Grupa DMP Games w czerwcu zaprezentowała pierwsze dwie wersje demonstracyjne gry „Chaos Guns”. Tytuł zalicza się do kategorii gier zręcznościowych i stanowi połączenie amigowych hitów „The Chaos Engine”, „Alien Breed” i „Hired Guns”. Akcję obserwujemy z góry i może w niej brać udział do czterech graczy jednocześnie – zarówno ze sobą współpracując, jak i konkurując. Dema są w pełni grywalne i pozwalają zdobyć ogólne pojęcie o tym, jak wygląda zabawa, jak rozwiązano sterowanie itp., ale niestety nie są zoptymalizowane i na minimalnej konfiguracji są praktycznie niegrywalne, czego nie można powiedzieć o szybszych maszynach, gdzie można spróbować zmierzyć się z poznaniem drzemającego w grze potencjału. Pierwotnie termin wydania gry planowano na rok 2014, lecz został on przesunięty na 2015, kiedy to miała być dostępna finalna wersja gry do pobrania za darmo z sieci (choć autorzy nie wykluczali również specjalnej, limitowanej edycji pudełkowej z grą wydaną na krążku CD, z drukowaną dokumentacją, mapami siedmiu etapów, szkicami rysunków i tym podobnymi ciekawostkami). Niestety nad projektem zawisły czarne chmury. Zespół osób tworzących tytuł opuścił główny programista i trwają poszukiwania osoby lub osób, które mogłyby go zastąpić. Sprawę komplikuje nieco fakt, że nie może być to osoba z przypadku, lecz ktoś dobrze znający język PowerD, w którym gra powstaje. Z uwagi na okoliczności autorzy rozważają również kilka innych rozwiązań, jak choćby przepisanie tytułu do mniej egzotycznego języka, lecz jak na razie nie ma na ten temat żadnych pozytywnych informacji (choć próby trwają). Dostępne wersje demonstracyjne gry działają w pełni w zgodzie z systemem – wykorzystują karty graficzne oraz AHI. Minimalne wymagania to dowolna Amiga z chipsetem AGA, procesor 68030 oraz 16 MB pamięci. Ocena: 2 (za wolną, trochę przekombinowaną, ale z potencjałem, choć tylko trochę szkoda, że zapewne nic z tego dalej nie będzie).

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



Back to the Golden Age

Kolejny port gry z Atari ST przygotowany przez Meynafa. Gra jest bijatyką utrzymaną w konwencji „Golden Axe” z elementami gry przygodowo-zręcznościowo-RPG. Wędrujemy rycerzem, wycinamy w pień wszystkich napotkanych nieprzyjaciół, odwiedzamy domostwa, sklepy, karczmy, gdzie możemy znaleźć lub kupić rozmaite przedmioty, które wspomogą nas w trudnym zadaniu. Ogólnie tytuł nie wygląda źle i całkiem nieźle się w niego gra. Brakuje jednak jakiejś podstawowej dokumentacji czy prostego opisu funkcjonalności i rozwiązań zastosowanych w grze (trzeba przemierzać internet w nadziei na znalezienie pomocy). Tytuł do działania wymaga Amigi z procesorem 68020. Sterować można myszą lub klawiaturą. Grę można uruchomić w wersji francuskiej oraz angielskiej. Ocena: 3 (produkt skończony i nie wygląda na kolejnego amosowego, niedopracowanego potwora).

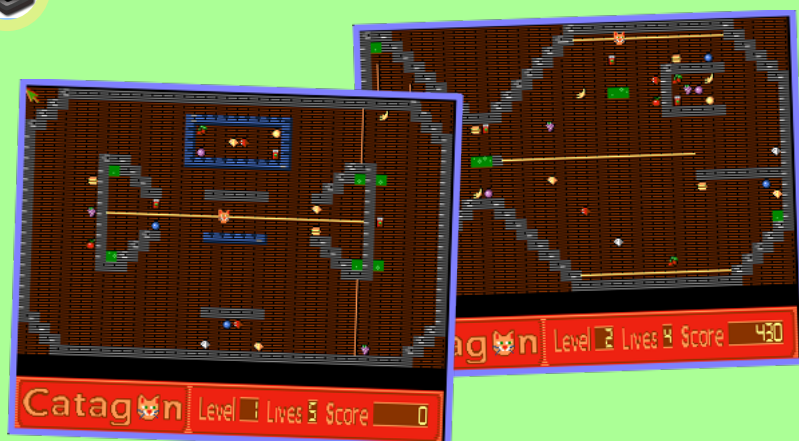
<http://meynaf.free.fr/pr/bttga.lzx>

Marco Gregorio's Adventures to Unusual Places

Kolejna pozycja stworzona przy wykorzystaniu silnika Backbone. Należy do gatunku platformówek. Gracz kieruje poczynaniami tytułowego bohatera i musi przebrnąć przez pięć etapów, a w każdym z nich odnaleźć wszystkie rozrzucone diamenty. Największą bolączką jest, jak to zwykle przy Backbone, animacja oraz detekcja kolizji, zwłaszcza że w grze trudno czasami rozróżnić co jest tem, a co nim nie jest. Pomijając to, tytuł wygląda całkiem przyzwoicie, choć trochę irytującą rozwiązano kwestie związane już z samą zabawą. Rozgrywka toczy się na rozbudowanych mapach, lecz brak jakiegokolwiek licznika czy nasz cel w zbieraniu diamentów jest osiągnięty sprawia, że nie mamy pewności czy „to już to”, czy też „znowu muszę wracać na sam początek”. Dodajmy do tego dużą liczbę ukrytych pomieszczeń, brak systemu kodów oraz zapisu stanu gry (gdy stracimy życie etap zaczynamy od początku i wszystko od początku musimy zbierać) i zamiast się cieszyć z rozbudowanej gry, przychodzi kubek dziegciu. Gra powinna działać na każdej Amidze z 2 MB pamięci. Ocena: 2,5 (za kompletność, wygląd i trud).

<http://eab.abime.net/showthread.php?t=73383>





Catagon

To amigowa odmiana gry „Paragon”. Gracz steruje, a raczej wpływa na ruchy będącego w ciągłym ruchu potwora, który musi zebrać wszystkie owoce i diamenty rozrzucone w danym etapie. Potworek odbija się od ścian pod kątem 45 stopni niczym kulka „Arkanoida” z tą różnicą, że jeżeli napotka na swojej drodze linę, przykleja się do niej, a gracz może wywołać jego wybiecie sterując w ten sposób miejscem odbicia. Gra na początku trochę chaotyczna, z czasem nieco irytująca, bo ileż można próbować trafić w jeden z elementów planszy. Graficznie – bez rewelacji, muzycznie nie najgorzej. Za całość odpowiada BSzili, który słynąc z wielu portów dla różnych systemów mógł się nieco bardziej postarać. Ocena: 2.

<http://eab.abime.net/showthread.php?t=75011>

Centron 3D

To nowa gra spod szyldu RPC-Games. Jest to strzelanina najprawdopodobniej stworzona w Amosie. Zadaniem gracza jest zniszczenie kolorowych kul przed upływem czasu, unikając min i statków Obcych. Gra składa się z 23 etapów i dystrybuowana jest na 2 dyskietkach DD. Zainteresowani mogą ją kupić za 12 dolarów australijskich dokonując płatności przez PayPal. Gra jest tak słaba, że nie wypada jej nawet publikować, o pobieraniu za nią pieniędzy nie wspominając. Graficznie i dźwiękowo jest to zrzynka z innych gier, z których autorzy po prostu wydłubali niektóre elementy i stworzyli swój własny kolaż. Unikać. Ocena: 1 (powinno być 0 – za wszystko).

http://www.rpcgames.com.au/Centron3D_Amiga.html



Donkey Downfall

Michael Gibbs podczas party Solskogen 2014 udostępnił dwie gry dla Amigi 500. Pierwsza z nich to nowa wersja recenzowanej już na naszym łamach „Flappy Bird”, w której autor skupił się na poprawkach i drobnych usprawnieniach. Druga pozycja to tytułowy „Donkey Downfall”. Jest to typowa gra zręcznościowa wzorowana na tytule znanym na konsoli Atari Jaguar. Zadaniem gracza jest przemieszczanie się postacią po platformach unikając zmiążdżenia przez górną część ekranu i upadku w nicłość. Pierwszą próbę przeniesienia tytułu na Amigę popiełnił całkiem niedawno Graham Humphrey (gra „Downfall”). Obie gry, pomimo że koncepcyjnie podobne, różnią się nieco sposobem zabawy. „Donkey Downfall” jest podzielona na etapy, podczas gdy „Downfall” oferował ciągłą rozgrywkę. Tytuł powstał w Amosie i jest przykładem tego, że w tym języku też można stworzyć dobrze wyglądającą grę. Program do działania wymaga Amigi z minimum 1 MB pamięci Chip. Ocena: 4 (przyjemny wygląd, udane wykonanie, całość może trochę nudzić, ale to już po prostu taka gra).

<http://www.pouet.net/prod.php?which=63736>

Renegades Deluxe

A to kolejna produkcja napisana w Amosie, której autorzy wstydzić się nie powinni. Gra jest co prawda jeszcze z poprzedniego wieku, lecz w październiku 2014 roku zdecydowano się wydać jej odświeżoną graficznie wersję z dopiskiem „deluxe”. Gatunek: strzelanina. Liczba graczy: dwóch. Cel: zniszczyć rywala. Gra się przyjemnie (zwłaszcza, że jest sporo opcji konfiguracyjnych), przyjemne wykonanie (do wyboru kilku wojaków, mnogość detali graficznych, lejąca się krew od zadawanych ran). Szkoda tylko, że poza głównymi bohaterami nie ma innych przeciwników i że nie można grać w pojedynkę (zawsze trzeba mieć kompana). Ocena: 3 (tytuł spokojnie zasługuje na 4, ale punkt niżej za brak opcji dla jednego gracza).

<http://wayneashworthart.com/Amiga.html>





Moim pierwszym komputerem była Amiga 500, pierwszą konsolą Pegasus MT-777DX (którego zresztą mam do dziś). W dzisiejszych czasach największym problemem Pegasusa jest podpięcie tej konsoli do nowoczesnego telewizora. Sprawia to niestety nie lada problem, a jakość grafiki... Cóż, wydawało mi się, że te gry były ładniejsze, chociaż może to wina telewizora, który nie lubi tak ekstremalnie niskich rozdzielczości – 256x240 pikseli na 40-calowym ekranie FullHD nie prezentuje się za pięknie. No ale nie wszystko stracone – istnieją przecież jeszcze emulatory. MorphOS oczekiwał się wersji 1.4 emulatora konsol NES/Famicon/Pegasus pod nazwą FCEU.

Instalacja FCEU nie jest skomplikowana. Pobieramy archiwum, a następnie wypakowujemy jego zawartość. W katalogu z FCEU, do katalogu ROMS wrzucamy wszystkie ROM-y z grami. Po uruchomieniu emulatora, pokazuje się główne okienko, gdzie możemy:

- ustawić sposób wyświetlania (Window, Overlay, Scanline, Vsync),
- skonfigurować podpięty Pad przyciskiem *Configure*,
- siłę dźwięku,
- wybrać z jakiego regionu są gry (PAL lub NTSC).



ginalnym Pegasusie podłączonym do telewizora LCD. Następnie zaznaczamy wybraną grę (pamiętajmy ROM-y muszą znajdować się w katalogu ROMS), naciskamy przycisk *Start* i można się bawić.

FCEU 1.4 testowałem na Macu MDD wyposażonym w PPC

G4 1.5 GHz z ATI Radeon 9600PRO 128 MB oraz na PowerBooku G4 1.5 GHz z ATI Radeon 9700M – obydwa komputery mają zainstalowany system MorphOS 3.7. W większości gier obciążenie CPU wynosiło

ok. 40-45%. Obecnie moja biblioteka gier to jakieś 50 tytułów i jak na razie tylko jeden ROM nie chce działać i jest to słynny kartridż dołączany do Pegasusa MT-777DX o nazwie „168-in-1”. Podczas gry możemy zapisać nasze postępy, nawet jeżeli pierwotnie gra na to nie pozwalała – wystarczy wcisnąć klawisz TAB i otwiera się menu, w którym mamy wybór zapisać grę lub odczytać wcześniej utworzony zapis. Gdy chcemy zakończyć grę, wystarczy nacisnąć klawisz ESC. Z ciekawostek należy zaznaczyć, że pod FCEU możemy bez najmniejszych problemów skonfigurować Pad z konsoli Playstation 4 pod warunkiem, że podepnijemy go kablem USB do komputera. Nie działają gałki analogowe oraz touchpad razem z przyciskami *Share* i *Options*, ale pozostałe klawisze można dowolnie konfigurować. Przykładową konfigurację klawiszy Pada można zobaczyć na zdjęciu.

Strona projektu:

<http://www.fceux.com>

Wersja dla systemu MorphOS:

<http://members.shaw.ca/realstar/>



Zatrzymam się chwilę przy opcjach wyświetlania, które mogą wymagać nieco więcej uwagi niż pozostałe. Jeżeli chcemy grać w oknie, to wybieramy *Window* (z automatu włączy się nam też *Overlay*). Natomiast jeżeli chcemy pograć na pełnym ekranie, to polecam *Overlay+Scanline* – obraz jest wtedy rozciągnięty na pełny ekran. Nie wygląda to może pięknie, ale na pewno zdecydowanie lepiej niż na oryginalnym Pegasusie.



Jeżeli ktoś lubi retro-gry, to emulator NES jest dla niego jak zbawienie. Podczas pisania tej recenzji podpiąłem „na chwilę” pad z PS4, aby uruchomić kilka gier i „zmarnowałem” prawie dwie godziny. „Dizzy”, „Big Nose the Caveman”, „Rygar”, „Micro Machines” – można długo wymieniać. Do zalet na pewno można zaliczyć bezproblemową współpracę z padem od Playstation 4, chociaż jakby nie patrzeć, to także zaleta MorphOS-a, który bezproblemowo go rozpoznał. Wad emulatora nie zauważyłem – działa stabilnie, nie przycina grafiki czy muzyki. Pomijając jeden przypadek, nie znalazłem jak na razie gry, która by nie działała na FCEU. A jeżeli chcielibyśmy na siłę znaleźć jakąś wadę, to potrzeba nam czasu. DUUŻO czasu, aby wszystkie te wspaniałe gry ograć ponownie.



Konfiguracja przycisków na padzie od Playstation 4

Aleksander „trOLLO” Giedyk



Rozmowa z **Tomaszem Golikiem** – właścicielem firmy **TimSoft**

Wielu z nas pamięta swego czasu funkcjonującego prężnie koszalińskiego wydawcę gier komputerowych na nasze ulubione platformy, jakimi są Commodore 64, Amiga oraz Atari. Polskie produkcje wydawane przez tę firmę na stałe wpisały się w pamięci komputerowców. TimSoft ma na swoim koncie takie hity, jak np. „Lazarus” czy „Kupiec”, nad którymi niejeden z nas spędził wiele godzin. Oto wywiad z właścicielem firmy, panem Tomaszem Golikiem:

Od kiedy zajmuje się Pan komputerami?

Zacząłem działalność pod koniec lat 80-tych, jako firma zajmująca się przeróbkami magnetofonów do Atari oraz powielaniem i sprzedażą gier na ten komputer. Już na początku działalności podpisałem umowę z chorzowską firmą Atares, na mocy której zyskałem prawo do instalacji w magnetofonach do Atari systemu Turbo Blizzard, który wielokrotnie zwiększał transfer danych z oraz do kasy.

Jaki był Pana pierwszy komputer?

Atari 800XL – magiczna maszyna kupiona za 110 USD w Peweksie.

Jak rozpoczęła się działalność TimSoft?

Zawsze marzyłem o pracy, do której wystarczyłoby biurko w domu. Interesowałem się trochę elektroniką i zafascynował mnie kolosalny wzrost transferu danych, jaki osiągały magnetofony do Atari po instalacji modułów Turbo. Nawiązałem kontakt z Ataresem i tak się zaczęło. Oczywiście kopiowałem wtedy także gry (było to w tamtych czasach legalne w Polsce). Na początku lat 90-tych, w odpowiedzi na zapotrzebowanie rynku, poszerzyłem ofertę o gry na Amigę i C-64.

Ile osób pracowało w firmie?

2 osoby + ja. Plus mnóstwo osób współpracujących, w szczególności grafik (od poligrafii) Adam Knasiecki.

Skąd pomysł na nazwę?

Przejdźmy do następnego pytania.

Jak zdobywano w Polsce wtedy rynek?

Początkowo działałem tylko lokalnie i wystarczała reklama szeptana wzmocniona poprzez drukowane przeze mnie regularnie katalogi gier (czarno-białe, ksero). Potem wprowadzono ustawę o prawie autorskim, która chroniła od tej pory także prawa zagranicznych wydawców gier. Jeszcze przez kilka lat można było sprzedawać pirackie gry, jednak sklepy zaczęły poszukiwać oryginalnego oprogramowania z troski o swój byt w nadchodzących czasach. Był to moment, kiedy nieliczne, dostępne na rynku oryginalne produkty stały się towarami poszukiwanymi. W 1992 roku wydałem „Ortotris” – grę edukacyjną na PC autorstwa Łukasza Knasieckiego (inspirowana znaną grą polegającą na odpowiednim układaniu spadających klocków – „Tetris”). Była ona jed-

nym z hitów mojej firmy. Po początkowym, próbnym okresie sprzedaży, gra doczekała się modnego w tamtych czasach wielkiego kartonowego opakowania z drogą poligrafią. To był jeden z przełomów w historii firmy – poważne sklepy komputerowe wymagały poważnie prezentujących się na półce produktów. W 1993 roku, we współpracy z lokalnymi dydaktykami, wydałem pierwszy oryginalny produkt na małe komputery: „Chemię”. Tytuł ten zapoczątkował całą serię pro gramów edukacyjnych, początkowo na komputery Atari, a w kolejnych latach na pozostałe dostępne platformy (C-64, Amiga i PC). Dzięki współpracy z koszalińskim grafikiem Adamem Knasieckim i szczecińską drukarnią moje produkty wyróżniały w tym okresie się na tle rynku doskonałą, nowoczesną poligrafią.

Czy było trudno walczyć o swoje miejsce na rynku?

Nie. Byłem jednym z pionierów i dzięki współpracy ze świetnym grafikiem oraz nowoczesną drukarnią mogłem dostarczyć na zdominowany przez postkomunistyczną szarość rynek atrakcyjnie opakowane, niezłej jakości produkt.

Czy rynek był wtedy chłonny?

Tak, ale tylko na miarę małych, kilkuosobowych firemek (jak moja). Dla dużych, polskich graczy jeszcze nie było miejsca (dzisiaj zresztą też już nie ma – sam kupuję od kilku lat tylko gry w formie elektronicznej, głównie na Steamie).

Jak wtedy wyglądała kwestia piractwa i jak to się odbijało na sprzedaży?

Rynek zdominowany był przez pirackie produkty, nie opłacało się więc tworzenie i wprowadzanie na rynek produktów wysokobudżetowych. Siłą rzeczy bazowałem (a także moi konkurenci) na twórczości pasjonatów – młodych ludzi, dla których sam fakt wydania ich gry był sposobem na zaistnienie na scenie, w przeciwieństwie do dem technologicznych dających niezłe (jak dla młodych ludzi) profity.

Jak wyglądał proces od zamówienia do wydania gry?

Trochę wstyd się przyznać, ale wtedy wymagałem od autorów produktów kolorowych, o chwytliwych (wg mnie) tytułach, często wzorowanych na produkcjach zachodnich. Po prostu rynek wymagał produktów podobnych do gier światowych (sprzedawanych jako pirackie kopie), jednak oryginalnych i nie przewyższających ich znacząco ceną.

Jaka wtedy była konkurencja?

Kilka niedużych firm (moja firma była wśród nich średnia) wyszukujących, tak jak ja, młodych autorów. Liderami byli L.K. Avalon i Mirage.



Jakieś niezrealizowane plany?

Raczej niewykorzystane możliwości. Był taki czas, że gracze w moim sklepie internetowym tworzyli bardzo aktywną grupę dyskusyjną. Nie potrafiłem tego wykorzystać (tzn. kreatywnie rozwijać narzędzi ułatwiających im korzystanie z forum) i poszli sobie gdzie indziej.

Jakie były dalsze losy firmy?

Kolejne lata przyniosły mi kilkadziesiąt tytułów gier i programów edukacyjnych na Atari, C-64, Amigę i PC. Strategia Timsoftu opierała się wtedy na wyszukiwaniu autorów, którzy stworzyliby dla nas edycje gier i programów na kolejne platformy. Dzięki temu mogłem opłacać naprawdę drogie akcje reklamowe – np. wrzutki w postaci opłaconych kart zamówieniowych w najpoczytniejszych branżowych periodykach – „Secret Service” czy „PC World Computer”. W połowie lat 90-tych podjąłem współpracę z firmą ASF z Gdańska, w efekcie której Timsoft wydał w 1995 roku grę „Miecze Valdgi II” na C-64 oraz Amigę. Pod koniec lat 90-tych, w miarę jak rosła oferta oryginalnych gier i programów dostępnych na polskim rynku, moja firma w coraz większym stopniu angażowała się w dystrybucję i sprzedaż detaliczną, wysyłkową. Wkrótce sklep wysyłkowy, tradycyjny, a po roku dwutysięczny sklep internetowy stał się dominującym rodzajem działalności. Działalność wydawniczą wygasilem. Łącznie przez kilkanaście lat TimSoft wprowadził na rynek około setki tytułów na wszystkie dostępne wówczas platformy.

Czy wraca Pan czasem myślami do tamtych czasów?

Tak. Kiedy gram w „Bioshock Infinite” lub w „Far Cry 3” uświadamiam sobie, że niebawem postęp graficzny, który dokonał się przez te 25 lat niewiele zmienił w odbiorze gier. Dokładnie tak samo jak dzisiaj dech zapiera grafika „Metro Last Light Redux”, tak wtedy „Barbarian” olśniewał wizualnie, a „Boulder Dash” wciągał intensywną i wciągającą rozgrywką.

Czy chciałby Pan coś przekazać obecnej społeczności pasjonatów komputeryzacji tamtych lat?

Cieszę się, że ktoś kolekcjonuje dzisiaj antyki, które były kiedyś dla mnie ważne.

Dziękuję za rozmowę.

Paweł Nowakowski



Synergy

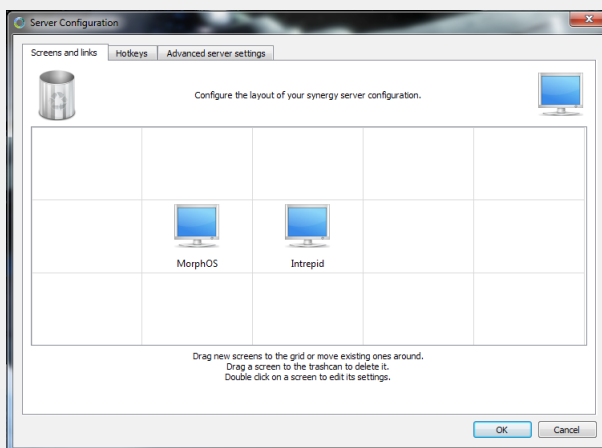
Co to jest *Synergy*? Jest to program „pchełka”, który pozwala na dwóch różnych komputerach korzystać z jednej myszki i klawiatury. Niby nic szczególnego, ale jakże jest to przydatne...

Jak to działa? Po pierwsze, potrzebujemy program główny tzw. serwer – w systemie MorphOS 3.7 jest tylko klient, który umożliwia podłączenie do serwera. W moim przypadku serwerem jest Synergy 1.5.0 w wersji dla Windows 7 64 bit. Instalacja nie jest skomplikowana, a konfiguracja jest równie prosta. Po uruchomieniu Synergy na PC, pojawia się okienko, na którym widać adres IP naszego komputera oraz możemy wybrać tryb pracy – w moim przypadku jest to wspomniany już serwer, który użycza myszy i klawiatury. Teraz przychodzi pora na konfigurację serwera. Jest to czynność niesamowicie prosta – wręcz banalna. Wystarczy wybrać opcję „configure server”, pojawi się nam okno z „mapą” komputerów. W prawym górnym rogu znajduje się ikonka monitora, którą należy przeciągnąć i umieścić w wolnym polu. W moim przypadku serwer nazywa się „Intrepid”, a komputer z systemem MorphOS – „MorphOS”. Nazwę klienta musimy sobie sami nadać.

Ułożenie ikon klienta i serwera jest bardzo ważne, ponieważ myszka przeskakuje na kolejny monitor w sposób, jaki ustawimy je w konfiguracji. Jeżeli mamy komputery skonfigurowane tak, jak to widać na zdjęciu, wtedy myszka przeskoczy na drugi monitor, gdy będzie znajdować się przy lewej krawędzi monitora serwera. Jeżeli zmienimy

ustawienia, to zmieni się też sposób zachowania myszy.

Gdy już zakończyliśmy konfigurację klienta pod Windowsem, przechodzimy do drugiego komputera, w naszym przypadku MorphOS-a. Uruchamiamy tam klienta Synergy. Wpisujemy adres IP serwera, port oraz nazwę – tę samą, którą nadaliśmy przy konfigurowaniu serwera (w moim przypadku, jak pamiętamy jest to „MorphOS”). Po konfiguracji klienta, uruchamiamy serwer na PC przyciskiem „Start”, a na komputerze kliencie wybieramy „Połącz”. Jeżeli wszystko przebiegło bez problemów, to klient wersji MorphOS wyświetli zie-



Konfiguracja serwera. Tutaj ustawiamy sposób współdzielenia myszy.



Klient połączony (zielona „żarówka”) i niepołączony (czerwona „żarówka”)

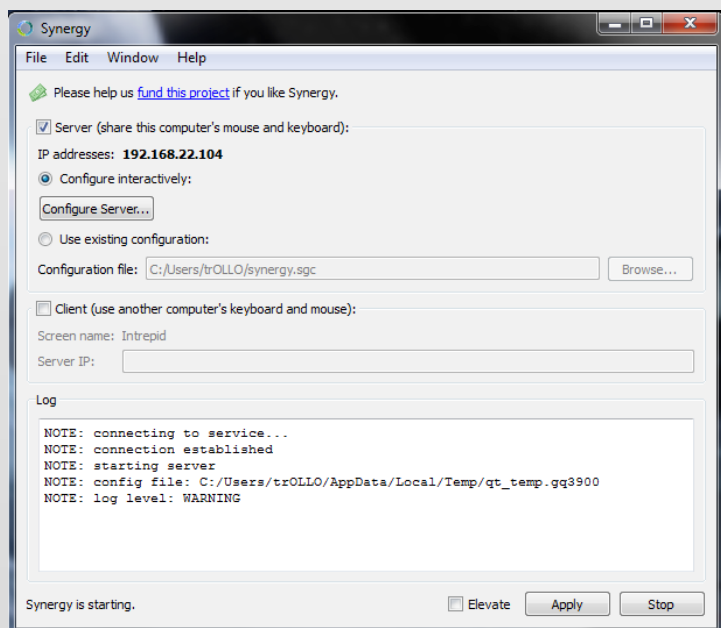
loną „żaróweczkę” i można rozpocząć zabawę.

Początkowo nie byłem przekonany do takiego rozwiązania – przecież każdy komputer ma swoją myszkę oraz klawiaturę. Jednak ciągle przeskakiwanie z jednej klawiatury na drugą jest po dłuższym czasie nieco uciążliwe. Synergy powoduje, że można bez problemu pracować na jednym ulubionym zestawie myszy i klawiatury. Dodatkowo, jeżeli mamy ustawione monitory jeden obok drugiego, to taka praca jest całkowicie naturalna – po prostu wskaźnik myszy przeskakuje na drugi ekran. Takie rozwiązanie ma też zalety, np. tekst pisany na jednym komputerze możemy wkleić na drugi, tak samo jest z przykładem z odnośnikami stron www. Szkoda natomiast, że w ten

sam sposób nie można przenosić plików między komputerami. Oczywiście nie ma programu bez wad. Zauważyłem, że na komputerze-kliencie nie można używać polskich znaków, co trochę czasami komplikuje pracę. Pomijając jednak tę niedogodność, obecnie, gdy już się przyzwyczaiłem, ciężko jest mi zrezygnować z Synergy.

Zastanawiać się można dla kogo jest Synergy? Na przykład dla takiej osoby jak ja, która ma mało miejsca na biurku, gdzie trzymanie dwóch dużych, wygodnych klawiatur jest w zasadzie niemożliwe. A przecież i tak w danym momencie użytkownika jednego z dwóch komputerów korzystamy tylko z jednej klawiatury i drugą zazwyczaj gdzieś odkładamy. Jest to wbrew pozorom ogromna oszczędność miejsca. Nie wspomnę już o tych osobach, które mogą mieć w domu po kilka komputerów i ciągnięcie tych wszystkich kabli od klawiatur i myszek lub choćby ich samych, gdy są bezprzewodowe, jest sporym utrudnieniem sobie życia. Synergy rozwiązuje je kilkoma prostymi kliknięciami.

Aleksander „trOLLO” Giedyk



Główne okno konfiguracji serwera

- jedna klawiatura i jedna mysz na dwa komputery,
- możliwość kopiowania tekstu pomiędzy połączonymi komputerami.

- brak polskich liter podczas pisania na komputerze-kliencie,
- brak możliwości kopiowania plików pomiędzy komputerami.



TALES OF GORLUTH

Szczerze mówiąc nie spodziewałem się, że w roku 2014 ukaże się komercyjna gra na klasyczną Amigę. Jak się jednak okazało, jest to możliwe i firma (grupa?) Amiworx Software, znana wcześniej z paru przeciętnych produkcji stworzonych w kreatorze Backbone, postanowiła uraczyć nas grą „Tales Of Gorruth: The Tear Stone Saga”. Szumne zapowiedzi obiecywały kolorową grę action-adventure w stylu słynnej „Zelda” z konsoli Super Nintendo. Czy zostały one spełnione? No cóż...

Zanim gra ukazała się oficjalnie, jej autor, Patrick Nevian regularnie udostępniał na forum EAB informacje o postępach prac oraz prezentował filmy demonstracyjne na Youtube. Już wtedy było wiadomo, że „Tales of Gorruth” jest kolejną produkcją stworzoną w amosowym gamemakerze Backbone. Jedyną pozytywną cechą Backbone jest możliwość stworzenia gry przez osoby, które nie potrafią programować. Pozostałe właściwości tego programu niestety nie powalają na kolana. Engine Backbone jest bardzo powolny, w praktyce zupełnie płynny przesuw ekranu nie jest możliwy, a produkcje są grywalne dopiero na Amigach z procesorem 68020 i pamięcią Fast. Dodatkowo rozliczne ograniczenia silnika, takie jak ubogie opcje sterowania czy wymuszanie liniowości fabuły powodują, że w mojej ocenie nadaje się on do stworzenia, co najwyżej, prostej platformówki.

Niezależnie jednak od wad samego silnika, oglądając filmy gry na Youtube, zauważyłem całkiem sympatyczną oprawę audiowizualną „Tales of Gorruth”, przypominającą gry action RPG na konsoli Super Nintendo czy Sega Megadrive. Dodatkowo moje zaciekawienie wzrosło, kiedy zobaczyłem fotki profesjonalnie wydanej gry na płycie CD z ładnym nadrukiem i instrukcją. Jako, że cena w sklepie Amiga-kit.com, wynosząca 14 euro nie wydała mi się aż tak wygórowana, postanowiłem złożyć za-

mówienie. Po kilku dniach oczekiwania otrzymałem przesyłkę zawierającą grę. „Tales Of Gorruth” wydana została na płycie CD-R, z profesjonalnym nadrukiem oraz okładką z krótką instrukcją. Całość sprawia naprawdę pozytywne wrażenie.

Instalacja i uruchomienie gry przebiega w sposób dosyć niestandardowy. Z płyty CD na twarde dyski należy przegrać katalog z grą, czyli w zależności od wybranej wersji językowej (angielski lub niemiecki) TOG-E lub TOG-D. Folder z grą musi się znaleźć bezpośrednio w głównym katalogu na dysku, a grę należy uruchamiać z AmigaDOS po uruchomieniu komputera bez Startup-Sequence. Niestety, jeśli uruchamiamy grę inaczej, może ona nie działać poprawnie, np. nie odczytują się kolejne epizody. Gra zajmuje około 36 MB na dysku twarde, wymaga 2 MB pamięci Chip, 1 MB Fast, zalecany jest też procesor 68030, chociaż na mojej A600 z 68020/28 MHz gra chodzi znośnie.

Po uruchomieniu gry możemy zapoznać się z krótkim intro wprowadzającym w fabułę. Główny bohater „Tales of Gorruth”, mistrzowski władający mieczem młodzieniec o imieniu Karn, zamieszkuje w krainie Middle Land, przez którą 45 lat wcześniej przetoczyła się wojna elfów z ludźmi. Szalony król elfów powołał swój lud na wojnę, w wyniku której doszło do eksterminacji elfów przez ludzi. Karn wraca do swojej rodzinnej wioski po latach nieobecności, tylko po to, żeby zostać włączonym w wydarzenia, które zmuszą go do bohaterskich czynów, sprowadzających się głównie do celnego rzucania (tak, tak, rzucania!) mieczem. Z intrygi dowiadujemy się też, że autorem scenariusza i muzyki w grze jest Patrick Nevian, a za grafikę odpowiada Javier Alcanis. Ciekawostką jest fakt, że oprawa graficzna nie została stworzona specjalnie na potrzeby „Tales Of Gorruth” i pochodzi ona z wydanej na telefon komórkowy gry „Dark Dimension” firmy FDG Entertainment. Zakładam, że autor miał zgodę na wykorzystanie grafiki z innej gry, bo logo FDG Entertainment pojawia się w intryze.

Obsługa w dosyć przejrzysty sposób opisana jest w książeczce dołączonej do płyty CD. Od razu rzuca się w oczy, że opcji sterowania nie ma zbyt wiele. Poza kierowaniem pos-



tać ruchami joysticka, jedyną opcją jest przyciskanie fire, które odpowiada za wszystkie możliwe czynności w grze, czyli atak, badanie otoczenia, rozmowę, otwieranie drzwi. Generalnie czynność, jaką podejmujemy przyciskając fire, jest zależna od tego czy mamy przed sobą wroga, postać, z którą możemy rozmawiać, drzwi czy np. skrzynię. Ciekawą opcją jest możliwość cheatowania w bardzo prosty sposób: przyciskając „1” na klawiaturze numerycznej otrzymujemy pełną energię, a pod klawiszem „2” możliwość przejścia do następnego etapu gry. Zapewne dodanie takiej funkcji podyktowane jest faktem, że nie ma możliwości zapisu stanu gry (kolejne ograniczenie Backbone). Martwi natomiast fakt, że uruchamiając produkcję na Amidze 600, nie mamy możliwości skorzystania z tej funkcji.

Po rozpoczęciu rozgrywki ukazuje się nazwa rozpoczynającego grę etapu – „Turmoil in Middle Land, Part 1” i po chwili przechodzimy do właściwej rozgrywki. Od razu rzuca się w oczy dość słaba animacja postaci, co raz w zderzeniu z całkiem niezłą grafiką statyczną. Pole gry zajmuje około 2/3 ekranu i aż prosi się o to, żeby zostało powiększone. Mimo tego, że ekran do największych nie należy, okazuje się, że dla silnika Backbone płynny przesuw tła jest nieosiągalny, w wyniku czego gra sprawia wrażenie dosyć słamazarnej. Nasz bohater człapie więc sobie niespiesznie, dzierżąc cały czas miecz i tarczę. Od razu zwróciłem uwagę na to, że po naciśnięciu fire Karn zamiast zadać cios mieczem, wystrzeliwuje go z okolic swojej klatki piersiowej. Jest to zupełnie surrealistyczne, bo po pierwsze nie ma limitu mieczy do wystrzelenia, a po drugie postać cały czas trzyma swoją broń białą w ręce. No dobra, może czeplię się detali, w końcu kraina gry jest magiczna i wszystko jest tam możliwe, ale jakiś niesmak pozostaje. U góry ekranu widzimy dwa paski, obrazujące energię postaci oraz liczbę żyć.

Pierwszą lokacją, jaką poznajemy, jest rodzin-



Nasz bohater w akcji



na wieś głównego bohatera, w której możemy porozmawiać z kilkoma ważnymi postaciami, takimi jak ojciec Kama i jego ukochana z czasów szkolnych. Słowo rozmowa jest niestety użyte trochę na wyrost, bo w praktyce sprowadza się ona do czytania kolejnych ekranów monologu, podczas którego nie mamy żadnych możliwości interakcji z postacią, którą „zagadaliśmy”. Najprawdopodobniej jest to kolejne z ograniczeń gamemakera Backbone. Szkoda, bo fabuła jest dość ciekawa i potencjalnie gra mogłaby być o wiele lepsza. Po odbyciu paru rozmów wprowadzających nas w historię udajemy się do lochu, w którym staczamy nasze pierwsze walki, a naszymi przeciwnikami są nietoperze i kościotrupy. Sama walka nie jest skomplikowana, jednak wymaga trochę wprawy, chociażby dlatego, że w styku z przeciwnikiem bardzo szybko tracimy energię, a wrogowie bardzo często pojawiają się blisko nas i jedynym wyjściem jest szybki zwrot i po uzyskaniu większego dystansu ostrzelenie przeciwnika z większej odległości. Elementem, który mnie zaskoczył okazały się teleportery w postaci... beczek, do których należało podejść i przycisnąć fire. W lochach pojawił się też pierwszy power-up w postaci szkatułki z energią. Po zwiedzeniu lochów i stoczeniu parunastu walk wychodzimy na zewnątrz, zapoznajemy się z kolejnym fragmentem historii i ruszamy w dalszą podróż. Tak jak podano na okładce, fabuła gry jest liniowa i nie mamy wpływu na to, jak się potoczy. Nie jest to samo w sobie wadą, ponieważ dokładnie

tak samo wygląda konstrukcja fabularna chociażby w seriach „Zelda” czy „Final Fantasy”. Kolejną lokację w grze stanowi las, najeżony przeciwnikami i trującymi krzakami, po pokonaniu którego odkrywamy kolejny element fabuły, czekają nas kolejne walki, zdobywanie lepszych broni (nadal rzucały mieczami) i tak dalej aż do zakończenia epizodu. Aby rozpocząć kolejną część gry, wybieramy opcję *Continue Quest* i w ten sposób kontynuujemy „*Turmoil in Middle Land, Part 2*”, wędrując po mapie świata do kolejnej lokacji.

Muszę przyznać, że mimo ograniczeń wynikających z zas-

tosowania Backbone, w „*Tales Of Gorluth*” gra się znośnie, w czym dużą rolę pełni bardzo dobra oprawa muzyczna i niezła grafika. Niestety wniosek z tego taki, że gra mogłaby być o dwie klasy lepsza, gdyby Partick Nevian zamiast korzystać z ulomnego gamemakera, znalazł programistę, który stworzyłby dedykowany specjalnie dla gry silnik. Nie trudno wyobrazić sobie „*Tales of Gorluth*” jako grę kompletną, nawiązującą do takich produkcji jak „*Zelda*” czy amigowe „*Legends*”.

Czy opłaca się „*Tales of Gorluth*” zakupić? Zależy to od tego czy gracz jest w stanie przymknąć oko na pewne niedociągnięcia, które opisałem. Na plus należy zaliczyć chociażby ładne wydanie gry, niezłą oprawę muzyczną, ładną grafikę i ciekawą fabułę. Główne minusy to ograniczone jak na grę przygodową, aspirującą do prostego RPG opcje interakcji z otoczeniem, brak rozbudowanych statystyk postaci, słaba animacja i skokowy przesuw tła. Grywalność w tym wszystkim jest po środku - na pewno nie odrzuca, jednak nie ma też zdecydowanej zachęty, żeby wracać do gry. Pozostaje mieć nadzieję, że sprzedane egzemplarze „*Tales Of Gorluth*” zdopingują Amiwork do znalezienia kodera, który napisze porządny silnik do sequelu gry. Jako ciekawostkę można odnotować fakt, że gra jest dostępna w nowym online'owym sklepie AmiStore, stworzonym przez A-EON.

Mateusz „twardy” Eckert

Oficjalna strona projektu:
<http://www.talesofgorluth.de>

Minimalne wymagania:

- Amiga z chipsetem ECS
- procesor 68020 (zalecany 68030)
- 2 MB pamięci Chip i 1 MB pamięci Fast
- twardy dysk
- napęd CD-ROM

Cechy „*Tales of Gorluth*”:

- Linearna gra w stylu action-adventure, zapewniająca około 5 godzin rozrywki,
- 40 specjalnie napisanych modułów muzycznych, grafika w 64 kolorach,
- angielska i niemiecka wersja językowa,
- gra podzielona na 3 odrębne przygody.



- estetyczne wydanie na płycie CD, z profesjonalną wkładką,
- niezła oprawa graficzno-muzyczna,
- dobrze napisany scenariusz.



- wykorzystanie Backbone,
- problemy z działaniem na nowszych, niż Amiga OS 3.1 systemach operacyjnych,
- konieczność uruchamiania gry z CLI, bez start-up sequence i nagrania jej w konkretnym katalogu dysku,
- ograniczone możliwości interakcji w świecie gry,
- powolność,
- brak możliwości zapisu stanu gry.

TALES OF GORLUTH



Profesjonalne wydanie gry



Nasza pierwsza gra

Kurs programowania z wykorzystaniem Visual Studio 2005 Express Edition i VBCC - część 9

W dziewiątym, ostatnim odcinku cyklu, zajmujemy się edytorem, który to umożliwi nam tworzenie plansz. Dodamy także największy możliwy wynik, czyli tabelę najlepszych wyników.

Pierwsza kwestia nasuwająca się na myśl w sprawie edytora: czy ma być osobnym programem, czy też powinniśmy go wbudować w grę. My skupimy się na tym drugim podejściu. Zatem dodamy odpowiedni wpis w menu tak, aby użytkownik widział, że jest dostępny edytor i po kliknięciu nastąpi przejście do edycji etapów. Tutaj powinno się wybrać jedną z możliwych kostek – a jest ich kilka – i za pomocą myszki nanieść w odpowiednie miejsce. Oczywiście musi być możliwy w jakiś sposób wybór planszy i, najważniejsza opcja, zapisanie tak powstałej planszy gry. Wygląda na nie-dużo, lecz niestety czeka nas sporo pracy, bo to co napisałem jest bardzo ogólne i dosyć mgliste. Warto na przykład zadać sobie pytanie gdzie owe kostki, które ma wybierać użytkownik, będą umieszczone. Na wszystkie te pytania mam nadzieję uda mi się odpowiedzieć.

Jeżeli chodzi o kostki, to umieścimy je w osobnym okienku, które będziemy musieli otworzyć, obsłużyć sygnały pochodzące od niego i zamknąć, gdy już skończymy edycję i zechcemy przejść do ekranu tytułowego. Wyobraźmy sobie, że już jest zrobione okno edytora, teraz chcemy zająć się obsługą sygnałów. Jak pamiętamy to w **boxo.c** jest główny rdzeń gry zawarty w funkcji *loop* i tam też umieścimy kod obsługi, kładąc go przed wywołaniem *Wait*.

```
if (g_beditMode)
{
    sigSet |= g_nEditorWinSignal;
}
```

Jak widzimy, nie ma tu nic magicznego. Nawiasem mówiąc, zmienną *g_beditMode* umieszczamy w module *boxo*, dodajemy za pomocą operatora alternatywy bitowej numer sygnału do wcześniejszych. Jasne jest, że dodajemy go tylko, gdy warunek jest spełniony, co będzie równoważne z otwartym już oknem edytora tak, aby istniał sygnał dla tego okna. Nie możemy bezpośrednio dodać tego numeru, tak jak poprzednie, gdyż okno edytora może jeszcze wtedy nie istnieć i, jak dobrze pamiętam podczas testowania, to gra zawisnie. Sama obsługa sygnałów pochodzących od edytora wygląda podobnie jak w przypadku dema gry.

```
if (g_beditMode)
{
    if (signals & g_nEditorWinSignal)
    {
        g_beditMode = signalsEditorWindow();
    }
}
```

Dodam tylko, że gdy *signalsEditorWindow* zwróci *FALSE*, to nastąpi zamknięcie edytora. Zanim przejdziemy do modułu *editor*, opiszę zmiany, jakie dotknęły moduł *tile*, gdzie trzymane są procedury rysujące po oknie gry kostki, napisy i znaki. Aby możliwe było rysowanie po innym oknie, możemy każdą funkcję

zaopatrzyć w dodatkowy parametr *struct RastPort* pRPort*. Wystarczy wtedy zmienić wszystkie aktualne wywołania tych procedur. Istnieje też inny sposób – trochę bardziej skomplikowany. Zamiast jednej funkcji *PrintChar*, zrobimy dwie – jedną dla gry (*PrintGameChar*) a druga dla edytora (*PrintEditChar*). Do tego będziemy potrzebować dwóch wskaźników na strukturę *RastPort* (jedna dla gry, a druga dla edytora). Niewątpliwym minusem, poza powtórzeniem funkcji, będzie konieczność ustawiania tych wskaźników i ten ciężar udźwigną moduły *window* i *editor*. Pozostaje jeszcze dodanie wpisu „editor” do menu w *title*, który uruchomi edytor, za pomocą znanego nam wskaźnika na funkcję.

W końcu dotarliśmy do sedna, czyli do modułu *editor*. Tutaj, podobnie jak w poprzednich odcinkach, dodajemy go w typowy sposób – dwa nowe pliki i aktualizacja *makefile*, nie zapominając go umieścić w *initTable* zaraz po inicjacji konfiguracji. Kod inicjujący i kończący są niewielkie. Pierwszy z nich nic nie robi poza zwróceniem *RT_OK*, a drugi ustawia tryb edycji na *FALSE* i jeśli to możliwe, zamyka okno samego edytora. Przypatrzmy się bliżej funkcji uruchamiającej edytor (to właśnie wskaźnik na nią jest umieszczony w *title*).

```
void StartEditor(void)
{
    g_beditMode = TRUE;
    m_pEditwin = (struct Window*)OpenWindowTagList(NULL, m_editwinTags);

    if (NULL == m_pEditwin)
    {
        g_pFnc = &title;
        KillEditor();
        return;
    }

    g_nEditorWinSignal = 1L << m_pEditwin->UserPort->mp_SigBit;
    m_pRPedit = m_pEditwin->RPort;
    g_pEditorPort = m_pRPedit;

    showLevelToEdit();
    printEditor();

    m_nTile = TILE_EMPTY;
    m_bTileChanged = FALSE;
    m_nSaveCnt = 0;

    g_pFnc = &editorLoop;
}
```

Na początku ustawiamy tryb edycji na *TRUE*, dzięki temu w pętli głównej w *boxo* będziemy mogli odbierać i analizować sygnał dla edytora. Dalej następuje otwarcie okna, gdzie umieścimy kostki z gry i krótką informację z pomocą, mówiącą między innymi, jakich klawiszy można używać. W przypadku niepowodzenia przy otwieraniu okna, powracamy do obrazka tytułowego. Następnie ustawiamy numer sygnału potrzebny przy komunikacji z funkcją *loop*, a wskaźnik na strukturę *RastPort* używaną w oknie edytora zapamiętujemy dla funkcji rysujących po nim. Metoda *showLevelToEdit* pokazuje planszę do edycji w oknie gry, bo tam będziemy ją zmieniać a *printEditor* ma za zadanie narysować wszystkie rzeczy związane z oknem edycji, czyli po lewej stronie kostki a po prawej pomoc. Zmienna *m_nTile* przechowuje aktualny numer kostki będący w edycji i na starcie będzie to pusty klocek. Zadaniem *m_bTileChanged*, będzie poinformowanie funkcji rysujących kostkę, że użytkownik zmienił ją na inną. Ostatnia zmienna *m_nSaveCnt* będzie używana podczas zapisywania etapów. Na końcu znanym nam sposobem, przecho-

dzimy do pętli głównej edytora. Z opisu funkcji *StartEditor* powoli rysuje się obraz, jak edytor mniej więcej będzie działał. Odrobinę go jednak rozszerzę. Jak wiemy będą dwa okna: pierwsze to okno gry, na którym będziemy widzieć planszę i za pomocą myszki będziemy

```
static void editorLoop(void)
{
    checkExitFromEditor();
    checkMouseInEditor();
    checkKeysInEditor();
    checkMouseInGame();
    changeEditTile();
    animateEditorArrow();
}
```

nanosić kostki, tworząc tym samym nowy poziom. Dodatkowo, aby ułatwić życie twórcy, pod myszką będziemy

widzieć wybrany klocek, a klikając lewym przyciskiem myszy będziemy stawiać kostkę, a prawym ją usuwać (czyli stawiać kostkę pustą). Drugie okno to okno edytora. Tu będziemy mieć możliwość wybrania kostki i zobaczenia pomocy. Sam wybór kostki będzie możliwy na dwa sposoby: za pomocą klawiatury, używając klawiszy od 1 do 5 oraz za pomocą myszki, klikając zwykcyjowo na obrazek przedstawiający kostkę. Zapis poziomu będzie możliwy tylko za pomocą myszki. Aby użytkownik był pewny, że zapis się powiódł, ikonka przedstawiająca dyskietkę, na chwilę zmieni się i pojawi się na niej napis „ok”. W przeciwnym razie będzie to napis „bad”. Przejdźmy do głównej pętli edytora.

Nie będę podawał kodu dla wywoływanych funkcji, bo zajęło by to zbyt dużo miejsca, przedstawię za to dwie najciekawsze procedury: *checkKeysInEditor* i *changeEditTile*. Ujmując rzecz z grubsza to w tej funkcji zachodzą trzy rzeczy. Pierwsza i najważniejsza to procedury z przedrostkiem *check* – obsługa interfejsu użytkownika, czyli przetwarzanie zdarzeń pochodzących od zarówno od klawiatury, jak i myszki. Mamy tu trzy funkcje do obs-

ługi edytora i jedną obsługującą grę, bo przecież tam klikamy myszką stawiając odpowiednią kostkę. Spójrzmy na *checkKeysInEditor* (na następnej stronie).

W tej funkcji sprawdzamy klawisze i podejmujemy stosowną akcję, czyli dla klawiszy od 1-5 będzie to zmiana kostki, a w przypadku kursorów, zmiana poziomu. Aby ułatwić sobie zadanie i uczynić kod odrobinę bardziej przejrzystym, zamienięm ciągi instrukcji *if else* na pętle. Do tego celu potrzebowałem struktury składającej się z kodu klawisza, pozycji pionowej, potrzebnej, aby pokazać użytkownikowi, co aktualnie zostało wybrane i na ostatnim miejscu umieściłem numer kostki. W ferworze walki są to magiczne liczby, które powinno się zamienić na stałe z pliku *tile.h*, co pozostawiam czytelnikowi. Mając strukturę, tworzymy tablicę, aby w pętli się po niej przesuwać, co też jest robione pod koniec funkcji. W klasycznym sposób, czyli za pomocą „ifów”, sprawdzamy klawisze odpowiadające za kursory (lewy i prawy) i przy udziale warunku sprawdzającego czy czasem nie wyszliśmy poza ustaloną liczb-



```
static void checkKeysInEditor(void)
{
    struct KeyEdit
    {
        int key;
        int pos;
        int tile;
    };

    struct KeyEdit keyTable[] =
    {
        {KEY_1, EDIT_EMPTY_POS_Y, 0},
        {KEY_2, EDIT_WALL_POS_Y, 1},
        {KEY_3, EDIT_ROCK_POS_Y, 2},
        {KEY_4, EDIT_SHIP_POS_Y, 3},
        {KEY_5, EDIT_EXIT_POS_Y, 4},
    };

    if (g_bLeft)
    {
        g_bLeft = FALSE;
        if (g_nLvlNumber > 0)
        {
            g_nLvlNumber--;
            showLevelToEdit();
        }
    }
    else if (g_bRight)
    {
        g_bRight = FALSE;
        if (g_nLvlNumber < (g_nAmountOfLvl - 1))
        {
            g_nLvlNumber++;
            showLevelToEdit();
        }
    }

    for (int i = 0; i < ARRAY_SIZE(keyTable); ++i)
    {
        struct KeyEdit item = keyTable[i];
        if (g_bkeys[item.key])
        {
            PrintEditChar(' ', m_nPosX, m_nPosY,
                COLORTXT_NORMAL);
            m_bTileChanged = TRUE;
            m_nPosY = item.pos + 4;
            m_nTile = item.tile;
            break;
        }
    }
}
```

bę plansz, zmieniamy poziomy.

Najbardziej tajemnicza funkcja wywoływana w *editorLoop* to *changeEditTile*. Odpowiada ona za poprawne wyświetlenie kostki na ekranie gry, przy czym dopóki użytkownik nie naciśnie przycisku myszki, to poziom nie ulega zmianie. Poziom gry składa się z kostek 16x16 i w takiej siatce stawiamy też kostki. Zatem jeśli pobieramy pozycję myszki, to musimy ją odpowiednio „przyciąć”, aby funkcja stawiająca kostkę, wpasowała ją dokładnie tam, gdzie należy. Dlatego wykonujemy małą sztuczkę związaną z iloczynem bitowym – zamiast dzielenia całkowitego przez 16, mnożymy przez 16. Ten wybieg działa niestety tylko dla potęg dwójki, ale tak się szczęśliwie składa, że 16 nią jest. Czy to nie przypadek? Jasne, że nie. Wybierając szerokość i wysokość kostek powinniśmy kierować się potęgami dwójki. Wtedy możemy wykorzystać różne ciekawe ich własności (na przykład $2^n + 2^n = 2^{(n+1)}$ czy też przesunięcia bitowe w lewo czy w prawo zamiast dzielenia i mnożenia). To taka dygresja. Wracając jednak do funkcji *changeEditTile*. Po pobraniu pozycji, musimy sprawdzić czy mieścimy się w wymiarach poziomu. Jeśli nie, to wychodzimy z procedury. Ostatni warunek sprawdzający wykonujemy tylko wtedy, gdy nowa pozycja zmieniła się o przynajmniej jedną kostkę od starej pozycji chyba, że klocek został zmieniony przez użytkownika, to wtedy musimy go podmienić, tak by widział, że nas-

```
static void changeEditTile(void)
{
    const WORD nPosX = GetGameMouseX() & ~0x0f;
    if (nPosX < 0)
    {
        return;
    }
    const WORD nPosY = GetGameMouseY() & ~0x0f;
    if (nPosY < 0 || nPosY > ((g_nLvlHeight-1)*g_nTileWidth))
    {
        return;
    }
    BOOL bChanged = nPosX != m_nOldPosX || nPosY != m_nOldPosY || m_bTileChanged;
    if (bChanged)
    {
        m_bTileChanged = FALSE;
        UBYTE* pLvl = g_pAllLvl + g_nLvlNumber * (g_nLvlWidth*g_nLvlHeight);
        UBYTE* p = pLvl + m_nOldPosX / 16 + (m_nOldPosY / 16) * g_nLvlWidth;
        PutGameTile(m_nOldPosX, m_nOldPosY, *p);
        m_nOldPosX = nPosX;
        m_nOldPosY = nPosY;
        PutGameTile(m_nOldPosX, m_nOldPosY, m_nTile);
    }
}
```

stąpiła zmiana. Po spełnieniu warunku przywracamy kostkę ze starej pozycji i stawiamy wybraną przez użytkownika na nowej pozycji. Do pełni sukcesu pozostała do opisanie funkcja *animateEditorArrow*, która, jak łatwo się domyślić, animuje strzałkę w oknie edytora, informując, co aktualnie mamy wybrane. Oczywiście to nie wszystkie funkcje w *editor*, ale pozostałe są na tyle proste, że po tylu lekcjach czytelnik będzie w stanie je zrozumieć samodzielnie.

Nadszedł czas na dodanie tabeli najlepszych wyników. Oczywiście, aby go zapamiętać, zapiszemy go w postaci pliku o znajomo brzmiącej nazwie **highscore**. Sprawa z pozoru błaha, ale i tu czyha na nas parę niebezpieczeństw. Przede wszystkim brak wyżej wymienionego pliku przez przypadkowe czy to celowe skasowanie, niewłaściwa zawartość bądź długość pliku, uniemożliwiają tym samym poprawne wyświetlenie i walidację. Zaczynając od samego początku – najlepszy wynik musimy w jakiś sposób przechowywać – nazwijmy go *g_nHighscore*. Oczywiście jest, że musi być on w stanie przechowywać także maksymalną ilość punktów, którą można zdobyć w grze, a co za tym idzie obydwie zmienne zarówno *g_nScore*, jak i *g_nHighscore* muszą być tego samego typu. Najlepszym rozwiązaniem będzie umieszczenie najlepszego wyniku w osobnym module. Zwyczajowo dodajemy dwa nowe pliki **highscore.c** i **highscore.h** do naszego projektu, aktualizujemy zasady w *makefile* i dodajemy odpowiednią linię w *initTable* w pliku **boxo.c** tuż za inicjacją bibliotek. Sam moduł będzie bardzo skromny, bo będzie składał się tylko z dwóch funkcji: *InitHighscore*, *KillHighscore*. Tam też umieścimy zmienną *g_nHighscore*. Oto i ciało modułu *highscore*:

```
#include "highscore.h"
#include "fileIO.h"

int g_nHighscore;
static char* name = "highscore";

/*=====*/
int InitHighscore(void)
{
    g_nHighscore = 0;
    ReadFile(name, (UBYTE*)&g_nHighscore, sizeof(int));
    return RT_OK;
}

/*=====*/
void KillHighscore(void)
{
    WriteFile(name, (UBYTE*)&g_nHighscore, sizeof(int));
}

/*=====*/
```

Wydawać by się mogło, że w *InitHighscore* zerowanie zmiennej nie jest potrzebne, bo przecież zaraz później *ReadFile* wczyta do niej wartość z pliku. Nic bardziej mylnego. Wystarczy sobie zdać sprawę, że funkcja czytająca może nie zadziałać, gdyż plik nie istnieje, dlatego właśnie ustawiamy na zero zmienną *g_nHighscore*. Konstrukcja wskaźnik na adres zmiennej umożliwi procedurze *ReadFile* wczytanie wartości bezpośrednio w *g_nHighscore*. A trzeci parametr w wywołaniu funkcji oblicza rozmiar zmiennej. *KillHighscore* ma tylko jedno

zadanie – zapisać najlepszy wynik do pliku i to czynimy wykorzystując do tego celu *WriteFile*.

Oprócz modułu *highscore* trzeba zadbać jeszcze o prawidłowe działanie najlepszego wyniku w naszej grze. Pierwszym miejscem, gdzie powinna być logika z tym związa-

na, to miejsce, gdzie nowa gra się rozpoczyna, czyli u nas w funkcji *NewGame*. Zaraz za inicjalizacją początkowej ilości punktów dodajemy prosty warunek sprawdzający czy punkty nie są większe niż aktualny najlepszy wynik i jeśli tak jest, to wtedy *highscore* jest takie samo jak punkty. Drugim i ostatnim miejscem, gdzie powinno się sprawdzić punkty i najlepszy wynik to miejsce, kiedy gracz doszedł do wyjścia i uzyskał dodatkowe punkty. W tym przypadku postępujemy w taki sam sposób jak poprzednio. Na zakończenie warto wspomnieć, że wyświetlanie *highscore* znajduje się w *bottomPanel* w funkcji *UpdateHighscoreOnPanel*.

Myszę, że po dziewięciu odcinkach temat pierwszej gry w języku C został opisany, gdybym o czymś zapomniał a jest jeszcze wiele napisania, w ramach tego tematu, to proszę o informacja na forum PPA. Dziękuję i zachęcam do zadawania pytań oraz eksperymentowania z kodem.

Asman

Archiwum z pełną wersją kodu opisanej i rozpisanej na części pierwsze gry, którą autor nazwał roboczym tytułem Boxo, znaleźć można na naszym portalu.

<http://www.ppa.pl/artykuly/download/ftp/boxo9.zip>

Od redakcji:

Koniec cyklu programowania pod AmigaOS nie oznacza, że nasz periodyk nie będzie już przeznaczać więcej miejsca dla programistów. Wprost przeciwnie – od następnego numeru planujemy zaprezentować w tym miejscu coś dla bardziej zaawansowanych deweloperów, co jednak nie oznacza, że początkujący nie znajdą tutaj dla siebie również nic wartościowego.





Cave Story

Doukutsu Monogatari

Raz pewien Japończyk usiadł przed komputerem i zaczął klepać kod prostej zręcznościówki. Gość zwał się Daisuke Amaya (znany też pod ksywą Pixel), a swój niewinny projekt nazwał poetycko „*Doukutsu Monogatari*”. Jakies pięć lat później powiedział sobie: „*Joubu-n'na*”, co na nasze oznacza coś w stylu „*będzie tego*” i opublikował swoją pracę jako freeware. Ponieważ tytuł trochę trudno się wymawia (szczególnie amigowcowi – w końcu w nazwie jest ATARI), poza Japonią funkcjonował pod tytułem „*Cave Story*”, czyli opowieść z jaskini. Ciekawi co takiego można w byle jaskini zasłyszeć? To zapraszam dalej.

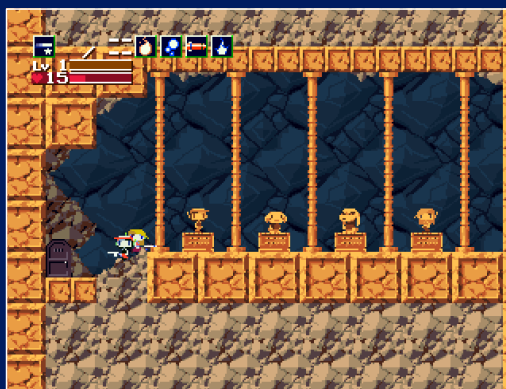
Odnosińniki:

<http://aminet.net/game/jump/cavestory.ppc-mos.lha>
<http://aminet.net/game/jump/cavestory.i386-aros.lha>

Po uruchomieniu gry naszym oczom ukaże się krótka animacja, przedstawiająca bliżej niezidentyfikowane osoby w dziwnym otoczeniu (później okaże się, że są to Doktor, Balrog i Misery, ale nie uprzedzajmy faktów), po czym będziemy mogli rozpocząć rozgrywkę. Zgodnie z tytułem, zaczyna się ona w dość ciemnej i obskurnej jaskini, a nasz bohater, ludek w czerwonych gatkach, bejsbolówce na głowie i antence w uchu nic nie wie i nic nie pamięta. Grafika, zarówno ta którą mieliśmy okazję zaobserwować w „Intrze”, jak i ta towarzysząca rozgrywce – mówiąc ogólnie – nie powala. Niska rozdzielczość, niewielka ilość kolorów, otoczenie może i schludne, ale spartańsko wykonane. Za to animacje zarówno postaci tytułowej, jak i przeciwników czy NPC-ów narysowane zostały przyzwoicie, wszystko porusza się naturalnie. W tle przygrywa (chyba, że uprzednio wyłączyliście to w menu) chiptunowa muzyczka (zmieniająca się w zależności od lokalizacji albo przy walce z bossami) – przyjemna, ale nie każdy przecież lubi chiptuny. Ogólnie rzecz biorąc – klimat retro pełną gębą.

„Ale o co w ogóle chodzi?”

– spyacie, spragnieni wyczerpujących informacji na temat kogo bić i za co. Otóż figę, panowie (i panie). Jak już wspomniałem, nasz ludzik wie tyle, co my. Dopiero w trakcie rozgrywki dowiadujemy się kim (czym?) jest, gdzie się znajduje, co ma zrobić i kim do licha są te gadające, królikopodobne stwory, które wszędzie się pałętają. Fabuła jest – myślę – na tyle ciekawa, że nie warto zdradzać szczegółów. Powiem tylko tyle, że grę można ukończyć na kilka sposobów – więc bardzo prawdopodobne, że jedno ukończenie Wam nie wystarczy i spróbujecie zaliczyć „maxa” (a przynajmniej tak było w moim przypadku), a dopiero to pozwoli Wam



poznać w całości zarówno historię świata, w którym się znajdujemy jak i naszego bohatera.

Rozgrywka – jak to w platformówkach bywa,

polega na chodzeniu, skakaniu po platformach i unikaniu (bądź eliminacji) grożących nam niebezpieczeństw. W grze znajdują się jednak dodatkowe, urozmaicające elementy w postaci: zbierania i korzystania ze znalezionych przedmiotów i wyposażenia, interakcji z napotkanymi osobami (jedno i drugie dostępne za pomocą strzałki w dół), jak również korzystania z inwentarza (dostępnego standardowo pod przyciskiem Q). Rozgrywkę zaczynamy z pustymi rękami, ale już wkrótce wpadnie w nie pierwszy (sfatygowany, ale sprawny) pistolet. Broni w grze będzie zresztą dość sporo, można mieć ich kilka jednocześnie i przełączać się między nimi w zależności od potrzeb czy okoliczności. Co ciekawe – w trakcie rozgrywki będziemy mieli możliwość bądź to zamiany, bądź przeróbki posiadanej giwery na inną, a to czy z niej skorzystamy czy nie będzie miało niemalże wpływ na dalszą grę. Zbierając wylatujące z przeciwników żółte trójkątki możemy podnieść aktualnie wykorzystywaną broń na wyższy poziom (co sygnalizowane jest żółtym paskiem po lewej stronie na górze ek-





ranu), dzięki któremu uzyska (najczęściej) większą moc, lepszy zasięg i/lub szybszą regenerację amunicji (z wyjątkiem rakiet – te trzeba po prostu zebrać). Aby nie było tak różowo, wyższy poziom broni łatwo jest stracić – wystarczy uszczerbek na zdrowiu (symbolizowanego z kolei czerwonym paskiem). Nasze siły vitalne podreperują z kolei bądź to wypadające z pokonanych przeciwników (czy napotkane w scenerii) serduszka (+2), bądź walające się tu i ówdzie regeneratory, mające postać pudełka z czerwonym serduszkem. Skorzystanie z nich uzupełnia zarówno nasze zdrowie, jak i zapas rakiet (o ile posiadamy wyrzutnię). W ostateczności można też przespać się na przygodnym łóżku – skutek jest taki sam, tylko bez rakiet w zanadru. Ciekawe, choć początkowo zaskakujące jest to, że każdorazowe wyjście z danej lokacji skutkuje odrodzeniem wszystkich pokonanych dotychczas przeciwników – wyobraźcie sobie, że przedarliście się przez tłum adwersarzy, na chwilę wchodzicie do jakiegoś domku, a po chwili – gdy go opuszczacie – voila! Wszystko po staremu! Całe szczęście, że mamy możliwość zapisania gry, przy pomocy rozrzuconych tu i ówdzie dyskietek. Wczytujemy ją potem albo z głównego menu (dowolny z pięciu dostępnych slotów), albo po kolejnej śmierci naszego bohatera (ostatni „sejw”).

Niestety, port gry dla MorphOS-a (być może również dla AROS-a, ale nic mi o tym nie wiadomo) posiada pewien okropnie irytujący błąd. Pod koniec Labiryntu czeka nas rozprawa z Rdzeniem (Core), a trzeba przyznać

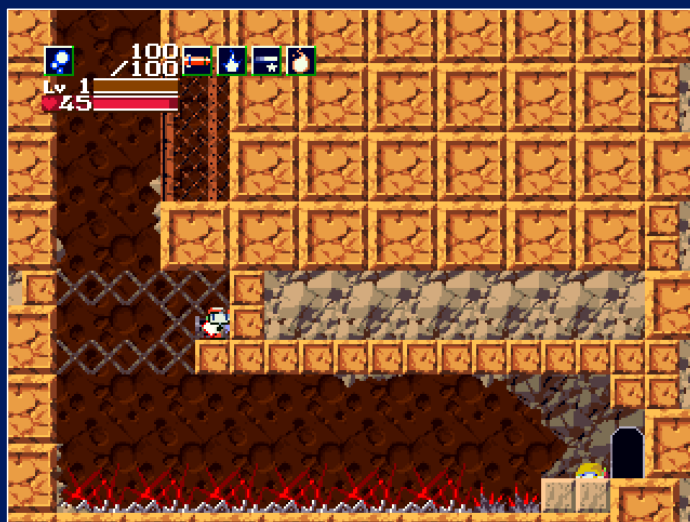


że jest to wyjątkowo upierdliwa walka. Na nasze nieszczęście, lokacja w której się ona rozgrywa ma skaszanione tło (jakiś kłopot z odświeżaniem – w efekcie „malują” nam się po

nim elementy pierwszego planu). Jak pewnie się domyślicie, wcale nie ułatwi to nam przebrnięcia przez to starcie, szczególnie jeśli przechodzimy grę na tzw. „fulla”. BSzilli twierdzi, że to jakiś bug w PowerSDL. Udało mi się zainteresować itixa problemem, więc trzymajcie kciuki, żeby zanim przeczytacie artykuł, pojawił się jakiś fix do gry, bo w przeciwnym razie będziecie zwalczać Rdzeń na niezłym hardcorze.

Nadeszła pora na małe podsumowanie. „Cave Story” to gra bardzo interesująca. Pomimo pewnego graficzno-dźwiękowego ubóstwa prezentuje wysoki poziom ogólnego dopieszczenia, interesujące detale, ciekawą fabułę i bardzo dobrze wyważony poziom trudności (nawet jeśli zginiemy w walce z jakimś bossem, to frustracja nie trwa długo i za jakiś czas robimy nowe podejście). I mimo, że gra nie ma „hade czyde”, „acziwmentów”, trybu online, czy filmowych wstawek z monumentalnym podkładem dźwiękowym, grało mi się w nią z prawdziwą przyjemnością i jestem pewien, że Wam również się spodoba - no, chyba że macie klaustrofobię.

Konrad Czuba



**AMIGA CD³²**

Full Motion Video

Jako kilkunastolatek, czytając gazety takie jak „Magazyn Amiga”, natrafiłem na informacje o module FMV dla Amigi CD32. Wówczas o takim urządzeniu mogłem jedynie pomarzyć. Moduł dla CD32 nie był jednak pierwszym tego typu rozwiązaniem. Jako pierwsza z owego rozwiązania korzystała konsola Philips CD-i, której producent wraz z koncernem Sony był także pomysłodawcą formatu VCD. Zarówno Philips jak i Commodore traktowały to urządzenie jako dodatek. Nie były one oferowane w zestawie z konsolą, a cena oscylowała w granicach zakupu samej konsoli.

Z technicznego punktu widzenia jest to sprzętowy dekodery MPEG-1 składający się z dwóch układów dekodujących: C-Cube – dekodery wideo oraz LSI-Logic – dekodery audio. Mimo, iż w specyfikacji ujęto możliwość odtwarzania płyt w formacie CD-i dotyczy to tylko bardzo trudno dostępnych filmów w tym formacie. O grach czy też płytach multimedialnych z konsoli Philipsa możemy zapomnieć, gdyż jest to całkowicie inna platforma sprzętowa.

Samo urządzenie montowane jest w złączu rozszerzeń specjalnie stworzonym właśnie na potrzeby modułu, co stało się później „furtką” dla firm trzecich w produkcji różnorodnych rozszerzeń przeobrażających Amigę CD32 w pełnosprawny komputer. FMV może być jednocześnie używane z modulem Paravision SX-1, jednak to rozwiązanie ma posiadać minusy – używając jednocześnie obu urządzeń mamy do dyspozycji 4 MB pamięci Fast, nawet jeśli w banku pamięci obsadzimy 8 MB. Co gorsze, moduł dużo bardziej się nagrzewa aniżeli w przypadku używania tylko FMV.

Głównym zadaniem tego modułu jest odtwarzanie filmów VCD. W standardzie PAL jest to



rozdzielczość 352x288 przy 25 klatkach na sekundę, a dla NTSC 352x240 przy 29,97 klatkach na sekundę. Przepustowość wideo i audio jest identyczna dla obu standardów i wynosi odpowiednio 1150 kb/s oraz 224 kb/s. Pozwala to na zapisanie na jednym krążku CD około 70 minut filmu. Jakość obrazu zbliżona jest do kasyety VHS, jednak zaletą pozostaje fakt, że jakość nagrania nie spada wraz z kolejnymi odtworzeniami. Moduł miał być także wykorzystywany w grach do odtwarzania sek-

wencji wideo, jednak poza „Cannon Fodder” żadna gra z niego nie korzysta, gdyż producentom znacznie łatwiej było przenosić gry na nośnik niż wzbogacać je o dodatkowe, czasochłonne tworzenie sekwencji wideo na potrzeby fabuły gry.

Do obsługi odtwarzacza wymagany jest oryginalny pad. Menu odtwarzacza wideo jest proste i przejrzyste. Zawiera informacje o numerze segmentu wideo, czasie jego trwania oraz długości całego filmu. Za pomocą przycisków na padzie możemy włączyć zapętlenie odtwarzania, włączyć losowe odtwarzanie. Po włączeniu odtwarzania można je wstrzymać i ponownie wznowić, wyświetlać informacje o czasie odtwarzania danego segmentu/filmu, czasie pozostałym do końca odtwarzania segmentu/filmu a także przewijać film.

Reasumując, obecnie kwestia ceny pozostaje drugoplanowa, gdyż głównym czynnikiem stała się mała dostępność urządzenia zwłaszcza komplet z oryginalnym pudełkiem i instrukcją co sprawia, że Full Motion Video jest łakomym kąskiem dla kolekcjonerów. Co ciekawe, nadal można zakupić tysiące tytułów filmowych w wersji VCD z czego zdecydowaną większość stanowią filmy dla dzieci.

Trudno znaleźć jakąkolwiek informację, jaka liczba sztuk tego modułu została wyprodukowana przez Commodore, ale sądząc po dostępności na portalach aukcyjnych można oszacować, że nie więcej niż 1000 sztuk.

Mirosław Arciszewski





A604n – rozszerzenie pamięci dla A600

Amiga 600 standardowo wyposażona jest w 1 MB pamięci Chip. Wystarczy to może do prostych gier, jednak gdy chcemy pograć w bardziej wymagające tytuły, szybko okazuje się, że jest to ilość niewystarczająca. Aby zaradzić tej sytuacji, firma Individual Computers wypuściła na rynek rozszerzenie pamięci Chip – A604n. Jest to nowsza i mniejsza wersja dostępnego wcześniej rozszerzenia A603.

Płytkę wykonaną jest bardzo profesjonalnie. W porównaniu do rozszerzenia A603, wersja 604n jest dużo mniejsza, przez co łatwiej ją zamontować „pod klapkę”. Sam montaż też nie jest skomplikowany. Rozszerzenie można zamontować na dwa sposoby. Pierwszy to tradycyjne montowanie „pod klapką”, drugi sposób wymaga rozkręcenia Amigi. Razem z rozszerzeniem dostajemy „riser” portu zegara, który przykręcamy do płyty Amigi 600 – tak, jak to widać na zdjęciu. Po zamontowaniu i przykręceniu rozszerzenia jest ono bardzo mocno i „na sztywno” zamontowane, przez co nie musimy się martwić, że podczas np. przenoszenia Amigi, coś się rozłączy.

Na rozszerzeniu znajdują się gniazda, które umożliwiają:

- montaż scandoublera Indivision ECS,
- montaż karty dźwiękowej Delfina DSP,
- montaż zegara czasu rzeczywistego,
- montaż kontrolera USB, na przykład Subway USB.

Podczas montażu scandoublera Indivision ECS nie musimy demontować stacji dysków, tak jak to miało miejsce w przypadku starszych wersji rozszerzenia. Jest to duży plus. Wadą jest jednak brak zamontowanego zegara RTC – dodatkowy koszt takiego urządzenia to około 90 zł.

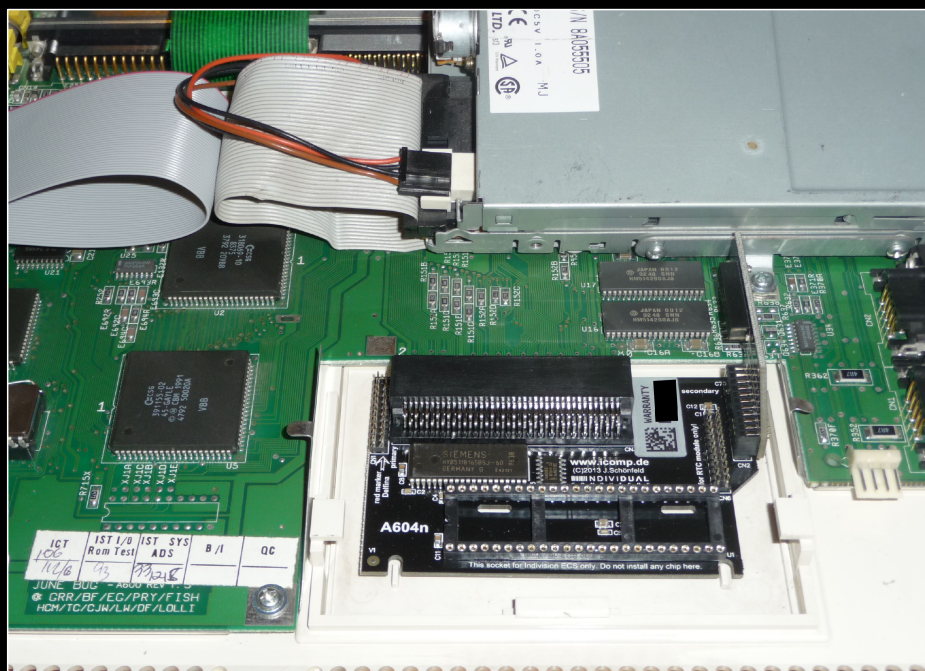


Urządzenie w działaniu spisuje się bardzo dobrze. Rozszerzenie pamięci Chip w A600 do 2 MB wbrew pozorom bardzo dużo daje. Na przykład w grze „Civilization” wyświetlają się wszystkie animacje, w „Franko – The Crazy Revenge” czy „Doman – Grzechy Ardana” wykrywana jest dodatkowa pamięć i rzadziej doczytują się dane z dyskietek. Przy takiej ilości pamięci można już z powodzeniem zamontować w Amidze 600 dysk twardy, zainstalować system i zacząć go wykorzystywać. Nabiera to wówczas sensu – po wczytaniu systemu, pozostaje jeszcze około 1.9 MB wolnej pamięci, z

którą można już coś sensownie zdziałać. Samo rozszerzenie nie wpływa na stabilność pracy Amigi – czterogodzinna „partyjka” w „Cywilizacji” nie była w stanie zawiesić Amigi.

Kartę kupiłem w RetroAmi za 145 zł i był to doskonały wybór. Część gier działa zdecydowanie lepiej, a 2 MB Chip pozwala na sensowną zabawę z Amigą 600 wyposażoną w dysk twardy. Rozszerzenie daje również możliwość instalacji bez większych problemów scandoublera Indivision ECS, karty dźwiękowej czy kontrolera USB.

Aleksander „trOLLO” Giedyk



- solidne mocowanie,
- możliwość instalacji Indivision ECS bez potrzeby demontażu wewnętrznej stacji dyskietek,
- możliwość montażu kontrolera Subway USB oraz karty dźwiękowej,
- cena.



- brak zegara RTC, który był obecny w poprzednich wersjach,
- dla niektórych wadą może być potrzeba rozkręcenia Amigi w celu montażu Risera clock portu.

RetroKomp 2014 - relacja

W dniu 8 listopada 2014 roku w salach budynku Gdańskiej Sceny Muzycznej, już po raz trzeci, odbyła się impreza RetroKomp. Tym razem tematem przewodnim zlotu były początki informatyzacji w Polsce, stąd w programie wykładów tematy takie, jak historia „Bajtki”, komputery Elwro, Meritum czy Unipolbrit oraz historia kultowego programu telewizyjnego z lat 80 – „Sonda”. Zwieńczeniem całodziennego festiwalu był koncert guru polskiej muzyki elektronicznej – Władysława Komendarka. Zabrakło niestety tym razem scenowego party LoadError (które według słów głównego organizatora, Sachego, jest planowane na rok 2015), jednak wątki demoscenowe były mocno akcentowane chociażby poprzez spotkanie z członkami legendarnej, commodorowsko-amigowej grupy Quartet oraz odbywającą się do późnych godzin „Noc z Demosceną”, podczas której prezentowane były na bigscreenie produkcje z różnych platform.

Dla mnie, jako jednego z organizatorów, impreza rozpoczęła się już w środę 5 listopada, kiedy to wraz z pozostałymi zaangażowanymi osobami rozpoczęliśmy przygotowania na miejscu, ustawiając stoły na sprzęt, przygotowując rzutniki itp. W piątek partyplac było w 100% przygotowane, a na miejscu było już wielu uczestników imprezy, w tym Tom Rain ze swoją imponującą kolekcją klasycznych Amig, Lotharek ze świeżym hardware wprost ze Śląska (w tym MIST z AGA-core), czy ARRGH!BYTE z czadowym zestawem dwóch A600 podłączonych do miksera, z których zapuszczał DJ-skie sety modułów z użyciem playera PT-1210 MK1. Impreza rozkręciła się na dobre, kiedy na miejscu pojawił się V-12 z członkami grupy Quartet: Polonusem, Jemasoftem i Hi-Manem. Panowie pojawili się na miejscu dokładnie w momencie,

kiedy prezentowałem na swojej Amidzie 1000 demo Quartetu „Christmas'89”. Od razu rozgorzała dyskusja o początkach sceny w Polsce, a ja nie omieszkalem poprosić chłopaków o autograf na dyskietce z ich demem sprzed 25 lat.

Oficjalny start imprezy, dla osób niezaangażowanych w organizację, miał miejsce w sobotę o 9 rano i od razu stało się jasne, że po raz kolejny RetroKomp cieszy się dużym zainteresowaniem. Imprezę tego dnia odwiedziło około



Publika dopisała



Dyskietko-obrazy

Tak jak napisałem na wstępie, w tym roku głównym tematem imprezy były szeroko pojęte początki polskiej informatyzacji. Temat ten przewijał się przez większą część wykładów, które spotkały się z bardzo dużym zainteresowaniem publiczności. Sala wykładowa była wypełniona szczególnie, począwszy od odczytu na temat wytworów polskiej myśli technicznej w postaci komputerów Unipolbrit, Meritum, Bosman i Elwro, poprzez wykład na temat historii czasopisma „Bajtek” (który prowadziłem wspólnie z Tom Rainem) czy prezentację legendarnej polskiej gry „Legion” przez Miro_1978 i wspomnienia Troya, autora jednej z pierwszych polskich gier na ZX Spectrum – „Boks”. Punktem kulminacyjnym okazała się prezentacja przypominająca popularno-naukowy program telewizyjny „Sonda”, który w latach

80-tych edukował społeczeństwo, w czasach siemieżnego komunizmu nie mające dostępu do nowinek technologicznych. Wykład prowadzony przez Sachego i Yolka uświetniła projekcja dwóch odcinków programu, specjalnie na ten cel wykupionych z archiwum TVP, dzięki sponsoringowi nieocenionego Sebastiana „Bachoo” Stecewicza z www.savethefloppy.com. Poza prezentacjami typowo wspominkowymi, uczestnicy RetroKomp mogli się zapoznać z grami „Castle Hack” i „Villages”, nad którymi pod szyldem Floating Coders pracują znani z forum PPA teh_KaiN i Proxy.

Po zakończeniu serii wykładów i podsumowaniu konkursu Łamaczy Joysticków, w głównej sali imprezy zaczęły się przygotowania do koncertu Władysława Komendarka, a większość uczestników udała się do sali wykładowej, w której rozpoczęło się spotkanie z członkami pierwszej polskiej grupy demoscenowej, Quartet. Niemal dwugodzinne spotkanie, prowadzone przez V-12/Tropyx i okraszone przy-



Tfardy i Tom Rain



gotowaną przez Johnyego prezentacją, spotkało się z ogromnym zainteresowaniem odwiedzających imprezę. Hi-Man, Jemasoft i Polonus, do których dołączyli ich przyjaciele i rywal z dawnych czasów, Silver Dream, swobodnie opowiadali o tym, jak zainteresowali się komputerami, jak powstały pierwsze polskie cracki do gier i pierwsze demo po naszej stronie żelaznej kurtyny. Bardzo interesujące okazały się także wspomnienia na temat gazety „64 Plus 4 & Amiga” oraz magazynu dyskowego „Kebab”, który przerodził się później w gazetę papierową pod tym samym tytułem. Spotkanie było filmowane i trwają starania o to, żeby udostępnić je w całości wszystkim zainteresowanym.

Kolejnym punktem imprezy był niemal dwugodzinny koncert Władysława Komendarka, poprzedzony pokazem filmu dokumentalnego na jego temat oraz rozmową publiczności z artystą. Po koncercie rozpoczęła się „Noc z Demosceną”, podczas której nastąpiła intensywna integracja retrokomputerowej i scenowej braci, w rytmie muzyki z dem pokazujących na bigscreenie. Nie zabrakło produkcji z takich platform, jak ZX Spectrum, Commodore 16/Plus4, Atari XL/XE, Atari Falcon oraz oczywiście Amigi. Pomiędzy demami swoje sety amigowych modułów zapodawał DJ ARRGH! BYTE. Impreza na dobre zakończyła się około trzeciej nad ranem, kiedy ostatnie osoby udały się na zasłużony odpoczynek do sleeping-romu.

Wygląda na to, że coroczne spotkania pod szyldem RetroKomp w Gdańsku stały się już „nową świecą tradycją”. Imprezę odwiedza coraz więcej gości, zarówno z Trójmiasta, jak i z innych zakątków Polski. Tegoroczne spotkanie relacjonowane było w ogólnopolskich stacjach TV (TVP, Superstacja), a mi się nawet udało pojawić w Teleekspresie. W roku 2015 ekipa RetroKomp planuje powrót do formuły „imprezy na podwójnym doładowaniu”, z nocnym demoparty i Retro Game Compo dla deweloperów. Plany są ambitne, więc trzymajcie kciuki i do zobaczenia za rok w Gdańsku!

Zapraszamy na stronę www.retrokomp.org i nasz profil na Facebooku.

Mateusz „tfardy” Eckert

Oficjalna strona spotkania:

<http://www.retrokomp.org>

Profil na Facebooku:

<http://www.facebook.com/RetroKompLoaderror>

PODZIĘKOWANIA

Przygotowanie takiej imprezy trwa ponad pół roku (dwuczęściowej jeszcze dłużej) i nie jest możliwe bez zaangażowania dużej, ale i zaufanej/sprawdzonej/odpowiedzialnej grupy osób. Wszyscy ci ludzie robią to po godzinach, w czasie nieraz „wynegocjowanym od rodziny” i to non-profit – niejednokrotnie inwestując własne środki sprzętowo-pieniężne, aby dany punkt programu zrealizować. Biorą też na siebie odpowiedzialność, aby z zadania się wywiązać na najwyższym poziomie (nie ma lipy), dlatego bardzo serdecznie im DZIĘKUJĘ:

- Ogromne podziękowania należą się Miastu Gdańsk, Scenie Muzycznej GAK oraz Tomkowi Maciukowi (SM-GAK) za całą pracę organizacyjno-promocyjną, infrastrukturalną i ogarnięcie finansów oraz całej koniecznej biurokracji. Bez Tomka nie byłoby imprezy;
- Wielkie dzięki dla Sebastiana „Bachoo” Stecewicza i inicjatywy www.SaveTheFloppy.com za sponsoring i wykupienie praw licencyjnych do materiałów TVP, które zostały wykorzystane we wspominkowym wykładzie o programie „Sonda”. Bachoo na moją prośbę bez wahania wsparł ten pomysł i tylko to umożliwiło jego realizację;
- Wielkie podziękowania dla gdańskiego Muzeum Starych Komputerów z Technikum Łączności oraz koszańskiego Muzeum Retro-elektroniki ElKom, za przygotowanie wykładów o wytworach polskiej myśli elektronicznej, jak i ekspozycje dużej (i ciężkiej) ilości eksponatów;
- Wielkie podziękowania dla grupy QUARTET (JemaSoft, Hi-Man, Polonus) oraz Silver Dreama – praojców sceny C64/Amiga, za przybycie i niezapomniane spotkanie z nami, pełne opowieści, anegdot i szczegółów technicznych oraz za przyjacielską postawę na całej imprezie;
- DJ'owi ARRGH!BYTE za urozmaicenie „Nocy z Demosceną” swoimi setami;
- Bartkowi Yolk – za przygotowanie i przeprowadzenie udanego wykładu o „Sondzie”, pomocy przy „Łamaczach Joysticków” jak i wsparciu przy realizacji „Nocy z Demosceną”;
- Tom Rainowi – za bardzo fajny wykład o „Bajtku”, pomocy przy „Łamaczach Joysticków” jak i wsparciu przy realizacji „Nocy z Demosceną”;
- „Tfardemu” – za współudział w wykładzie o „Bajtku”, pomocy przy prowadzeniu profilu FB jak i wsparciu przy realizacji „Nocy z Demosceną” oraz za całkowite wsparcie imprezy w wielu aspektach (tytuł że aż ciężko wszystko wymienić – dzięki!);
- Miro_1978 – za prezentację owianej legendą polskiej gry „Legion”;
- Markowi Trojanowiczowi (Troya) – za przygotowanie i przeprowadzenie wykładu o grze „Boks” (ZX, 1987);
- Floating Coders – za pokazanie postępów prac nad nowymi gramami na Amigę („Castle Hack” i „Villages”);
- Grey/MSB za wsparcie „Nocy z Demosceną” swoim cennym (i uruchamianym) Atari Falcon;
- V-12/TX – za przygotowanie merytoryczne i przeprowadzenie pogadanki z QRT;
- Drygol/LMS & Johnny/LMS – za wymyślenie i wyprodukowanie „RetroKOMP Star Award”;
- Sim1/WT+RSE+LMS+TTN – za oprawę graficzną strony www oraz logo imprezy;
- Mccnex/WT+LMS – za wsparcie i współorganizację całej imprezy, uruchomienie serwisu www, FB, wykonanie plakatu i dyplomów oraz niezliczoną ilość ważnych rzeczy;
- Cichy/Raybeez – za perfekcyjną robotę administracyjną www.serwer/forum RK;
- Johnny/LMS za współorganizację całej imprezy – poczynając od stworzenia materiałów wideo (rozszyfrowanie czołówki „Sondy”), przez prezentację-podkład do spotkania z grupą Quartet jak i nawiązanie kontaktu przekonanie do wystąpienia QRT na RK'14 oraz wzięcie na siebie całej masy różnych zadań organizacyjnych (również ciężkich zadań nocnych) - wielkie dzięki!

Oraz:

- Piter K., Traq, String/Agenda, At0m/LMS, Gepard/LMS, Szeryf/LMS, PitB, AdamK/LMS, Lotharek, Stryker – za różnorakie wsparcie i „kompaniję” (serdecznie przepraszam, jeśli ktoś pomógł, a ja to tutaj przeoczyłem – jednak bardzo dziękuję).

Dzięki wielkie wszystkim za udział (który też jest swoją pomocą dla nas, że to co robimy ma sens) i mam nadzieję – do zobaczenia jesienią przyszłego roku na:

RetroKomp/LOAD ERROR 2015

**Sachy/WT+RSE+LMS
(RK main org)**



Grupa Quartet



Zdjęcie rodzinne organizatorów



Riverwash 2014 Party Report

19 - 21 września 2014, Racibórz

Podczas gdy w Europie liczba imprez związanych z demosceną bezustannie rośnie (i wciąż pojawiają się nowe, również amigowe, inicjatywy), w Polsce – w tej materii – rok 2014 zapowiadał się nad wyraz skromnie. Zarówno łódzkie *WeCan*, jak i gdańskie *Load Error* (które co prawda się odbyło, ale bez części scenowej) zostały zawieszone. Honor uratowany został dopiero przez organizatorów *Riverwash*, które to po paroletniej przerwie powróciło na mapę imprez. Tegoroczna edycja „prania w rzece” miała się odbyć w Raciborzu, czyli na samym końcu świata, 10 km od granicy z Republiką Czeską.

Mimo, że *Riverwash* startowało oficjalnie w piątek 19 września (część nieoficjalna), to do Raciborza postanowiliśmy wyjechać w sobotę rano. Party jednak siłą rzeczy rozpoczęło się dla mnie i tak już w piątkowe popołudnie. Wtedy to w nasze wielkopolskie rejony „zrzucony” został *Jok/Dreamweb*, którego miałem za zadanie odebrać z miasta wikliny, czyli Nowego Tomyśla i przetransportować do naszej wolsztyńskiej bazy. Misja się udała, choć spontaniczne before party (do którego dołączył *Jazz-Cat*), przy kufkach złocistego napoju w przytulnym wolsztyńskim klubie *Locomotiva*, nie trwał za długo. *Jok* miał bowiem jeszcze do skoń-

czenia prace na komputery graficzne, a czas uciekał.

Następnego dnia rano wstaliśmy w doskonałych humorach, również dlatego, że prognozy pogody się nie sprawdziły i przywitała nas wyjątkowo ciepła (jak na tę porę roku), jesienna aura (a było to istotne z uwagi na położenie miejscówki, w której odbywała się impreza – o czym później). Spakowaliśmy niezbędny ekwipunek i wyruszyliśmy. Podróż w trójkę, którą umilała nam (tradycyjnie) demoscenowa ścieżka dźwiękowa, przebiegła spokojnie. Podczas jazdy, *Jok* nadal pracował nad swoimi grafikami, dzielnie walcząc z promieniami słońca, które to bezustannie atakowały ekran jego laptopa. Już koło 14:08 dotarliśmy do hotelu o szumnie brzmiącej nazwie „*Empire Hotel*”. Tam też zabukowałem kilka dni wcześniej dwa pokoje dla sześciu osób (reszta ekipy dojechać miała z innej części kraju). Niestety na miejscu okazało się, że w recepcji panuje totalny chaos. Wystąpiły „nieoczekiwane problemy” z naszą rezerwacją i wszystkie miejsca były już rzekomo zajęte. Pani recepcjonistka, o wyjątkowo srożej aparycji, była nieustępliwa i na nic zdały się groźby, że zmusimy ją do wysłuchania wszystkich utworów z „*ZX Spectrum Beeper Music Compo*” z rosyjskiego party

Chaos Constructions. Dopiero lęk przed odsluchaniem atarowskich „1-bit music competition” z ostatnich 10 lat okazał się silniejszy i pokoje szczęśliwie się odnalazły. Mimo, że hotel był imperium tylko z nazwy, nie mieliśmy powodów do narzekań. Czyste pokoje, ciepła woda (a to wcale nie jest takie oczywiste w polskich hotelach), niska cena z wliczonym dnia następnego śniadaniem (minimalistycznym i wymierzonym „co do plasterka”, ale jednak) i błoga odległość 500 metrów od party place były nie do przecenienia.

Czas wrócić do miejsca, w którym miało odbyć się party. Na niejednym scenowym evencie już w życiu byłem, ale takiej, co tu dużo pisać – genialnej – miejscówki, sobie nie przypominam. Otóż, *Panie i Panowie*, party *Riverwash 2014* odbyło się.... na Zamku Piastowskim. To nie wszystko. Zaraz „za ścianą” zamku, który robił za party place, usytuowany był najstarszy działający browar w Polsce, czyli Browar Zamkowy, produkujący doskonale piwo Raciborskie. Powiedźcie sami, czy można wyobrazić sobie lepszą lokalizację demoscenowej imprezy?

W miarę jak zbliżaliśmy się do bram zamku, coraz wyraźniej słychać było gromkie głosy



Część uczestników Riverwash 2014, autor: Argasek/Decree



wesołej gawiedzi. Okazało się, że trafiliśmy idealnie na jeden z (tych mniej oficjalnych) punktów programu *Riverwash*, czyli grupowe zwiedzanie sąsiedniego browaru (wliczone w cenę wejściówki). Orszak scenowców, prowadzony przez Ubika wyposażonego w megafon i wielkiego różowego lizaka (?) robił na – nieco zaniepokojonych – przechodniach piorunujące wrażenie. Po szybkich powitaniach od razu dołączyliśmy do peletonu, zostawiając formalności na później. Pani przewodnik szczegółowo i profesjonalnie opowiedziała zebrany o procesie warzenia piwa. Orowadziła nas po niewielkim, acz niezmiennie klimatycznym (ba, nawet ciut industrialno-steampunkowym) browarze. Zebrani z uwagą posłuchali zatem o zacieraniu śrutu w kadzi zaciernej tudzież schładzaniu brzezki gorącej w płytowym wymienniku ciepła. Nie muszę chyba dodawać, że wyprawa ta mocno nam wszystkim zaostrzyła pragnienie, które to – na szczęście – szybko mogliśmy ugasić przy ostatnim punkcie programu zwiedzania, czyli degustacji. Chętni mogli zakupić też wybrane rodzaje Raciborskiego w okazyjnych cenach. Duże brawa dla organizatorów za taką nietuzinkową atrakcję.

Party upływało w miłej atmosferze. Pogoda, o której wspominałem, była bardzo łaskawa i ciepły wrześnieowy weekend stanowił idealne tło do przesiadywania na dziedzińcu zamkowym, sączenia ulubionych napojów, konsumowania ulubionych potraw oraz długich rozmów ze starymi i nowymi znajomymi. Centrum imprezy stanowiła zamkowa restauracja, konkursy odbywały się natomiast w drugiej (połączonej korytarzem) sali (w swoim własnym wystroju). Dzięki wystawionemu na wesołość monitorowi, konkursy można było obserwować też wprost z dziedzińca. Konkurencje prowadzono sprawnie (co często jest bolączką na polskich imprezach), bez opóźnień i nieoczekiwanych incydentów. Moje uszy odnotowały wysoki poziom w obu konkursach muzycznych. Sporo prac i interesujący przekrój stylistyczny, a do tego sala słuchaczy pękająca w szwach (co jest ewenementem na naszych parties). W *Chip Music Compo* silny amigowy akcent zapewnili Aceman i JazzCat, prezentując doskonałe AHX-owe utwory. Kompoty graficzne również trzymały poziom. Zauważyłem kilka niezłych rysunków, a co ciekawe, coraz częściej uczestnikami graficznych zmagani są nie tylko panowie, ale i panie. Sprawne prowadzenie kompotów to z pewnością także i zasługa tego, że tegoroczny *Riverwash* został – z premedytacją – pozbawiony zbędnych koncertów, prezentacji sponsorów i całej tej niepo-



Zaczynają się konkursy, autor: Gorzyga/Decree

trzebnej otoczki, która tylko i wyłącznie przeszkadza i powoduje bałagan.

Przez RW przewinęło się około 100 osób. Rekordzistami byli członkowie grupy Lamers, którzy dojechali na miejsce w sobotę wieczorem już chyba jako ostatni partowicze. Jak powiedział mi at0m, na pomysł, aby jednak wlecieć na party wpadli spontanicznie na jakiejś imprezie dzień wcześniej, a że do pokonania mieli 600 kilometrów, to łatwo nie było. To się nazywa scenowy duch! Niestety nasza trójka (ja, JazzCat i Cahir) nie mogła oddać się w pełni owej beztroskiej atmosferze imprezy, ponieważ cały czas walczyliśmy z czasem, kończąc wspólnie produkcję, która wystartować miała w głównym compo.

Przed punktem kulminacyjnym każdego party, czyli konkursem na najlepsze demo, przeprowadzono jeszcze „Twoje stare demo kompo”. Ten oryginalny pomysł na – nieco luźniejszą – konkurencję zapożyczony został z *Xperience Party* (które to odbyło się w 2013 roku w Wałbrzyskim Ośrodku Kultury). Zasady są proste. Każda grupa może wystawić tutaj dowolne swoje demo z przeszłości. Warunek jest taki,

że nie mogą tutaj startować produkcje, które zajęły niegdyś pierwsze miejsca w konkursach – wybór najlepszego demo z najlepszych nie jest bowiem celem tego compo. Chodzi bardziej o to, aby skupić się na dziełach z drugich i dalszych miejsc, które to z różnych powodów nie zostały (zdaniem ich autorów) należycie docenione. Dodatkową atrakcją był wymóg, że dowolna osoba z ekipy tworzącej demo musi być obecna na party i przed projekcją wygłosić „przemówienie” na temat owej produkcji. „Twoje stare demo kompo” wypadło nad wyraz pozytywnie, zgłoszono aż 14 pozycji, a niektóre „przemówienia twórców” wywoływały spore salwy śmiechu wśród zgromadzonych.

W *Demo Compo* wystawiono 7 dem, co było niemalym zaskoczeniem. Spory aplauz wzbudziło „Timmy” grupy Specy.pl, wydane z okazji 30-lecia stworzenia komputera Timex 2048. Jest to najprawdopodobniej pierwsze na świecie demo, w pełni wykorzystujące możliwości graficzne tego – porośniętego już mchem historii – sprzętu. Dodatkowo, w przypadku uruchomienia dema na Timexie 2068, wykorzystywany jest wbudowany układ AY, na którym odgrywana jest muzyka. Produkcja, za którą



Demoparty na zamku? Czemu nie!, autor: Gorzyga/Decree



Wesoła wycieczka do browaru, autor: Gorzyga/Decree



odpowiedzialni są Tygrys (kod), graficy aT0m, Piesiu, i Ziutek (muzyka) przykuwa uwagę bardzo dobrym designem, zaawansowanym kodem, niezłą dynamiką i dobrej jakości oprawą graficzną. W dobie – jak to określił kolega Twardy „komputerowego ekumenizmu” warto odnotować, że jest to kolejne dzieło polskiej (stosunkowo nowej) grupy specy.pl stojące na światowym poziomie demosceny z pod znaku ZX.

Aż dwa dema na PC zaprezentowała grupa Futuris. Minimalistyczne w formie „CityGrid” (kod: asmCode, grafika: Biter, muzyka: Traymuss), z nostalgiczną ścieżką dźwiękową, przenosiło widza do odhumanizowanej, anonimowej metropolii. Produkcja druga, czyli „Burp” (kod: asmCode, grafika: Biter, muzyka: Traymuss) to z kolei bardzo zabawne demo z dużą dawką autoironii. Członkowie Futuris – śmiejąc się sami z siebie – nawiązywali w nim do swej wcześniejszej produkcji, która to, delikatnie mówiąc, „posiadała kilka drobnych mankamentów”. Mimo wszystko „Burp” to nie żart, a pełnokrwiste, dopracowane demo, z wysokiej jakości oprawą audiowizualną.

W konkursie na najlepsze demo pojawił się także akcent amigowy. Jako osoba czynnie zaangażowana w owy akcent, jego ocenę pozostawię innym, skupiając się tylko na suchych faktach. „Sunglasses at Night” sygnowana przez grupy Ghostown & Whelpz (kod: Cahir, grafika: Slayer, muzyka: JazzCat) to produkcja na Amigi OCS. Wersja zaprezentowana na party wymagała 1 MB pamięci Chip, choć – być może w chwili gdy czytacie te słowa – dostępna jest już wersja finalna, która odpali się na standardowej „pięćsetce” (i będzie zawierać szereg poprawek oraz tunin-

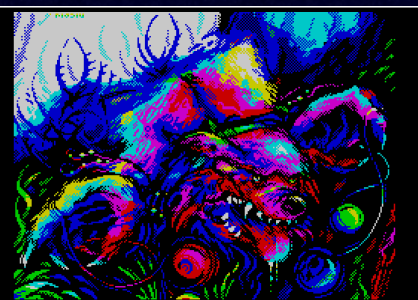
gów). tuningów). Cahir, który na party oficjalnie zasilil szeregi grupy Ghostown (co wiązało się – rzecz jasna – z odpowiednim obrzędem inicjacji, czyli koniecznością zjedzenia bez popijania dwóch odmian papryk z górnego przedziału skali Scoville’a, czyli papryki „Dorset Naga” i „Trinidad Scorpion Butch T”), po raz pierwszy zmierzył się z pisaniem „na małe Amigi”. Co prawda stwierdził później, że bardzo rozboleła go od tego pewna tylna część ciała, ale będzie pisał dalej. Co ważne, cały kod źródłowy naszego dema (napisany w języku C) został przez Cahira upubliczniony dla wszystkich chętnych (szczegółowy w napisach końcowych). „Sunglasses at Night” to demo utrzymane w nocnym, pulsującym i deep-housowym klimacie neonowego miasta, różniące się nieco od naszych wcześniejszych produkcji.



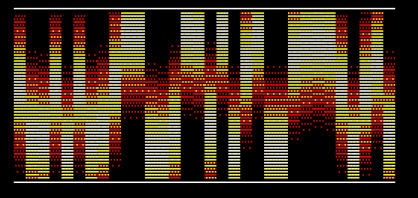
Screeny z dema „Sunglasses at Night” grup Ghostown + Whelpz (demo compo, drugie miejsce)

Nocny pokaz „Sunglasses at Night” zakończył sobotnie (a właściwie to już niedzielne) konkursy. Nie pozostało nam więc nic innego, jak założyć okulary przeciwsłoneczne i udać się na zasłużony chillout, nadrabiając integrację z partyzantami. Po raz pierwszy od wielu miesięcy nasz team Ghostown mógł się zebrać w pełnym składzie w jednym miejscu (Bodzio, Cosh, Codi, DKL – serdecznie pozdrawiam). Miło było spotkać dawno niewidzianych kumpli, pogadać – przy szklaneczce whisky – nie tylko o przeszłości, ale i o scenowych planach na przyszłość. Co ciekawe, po raz pierwszy od dawna, co najmniej kilka osób wspominało o demo planach związanych z Amigą – oby choć część z tych zapowiedzi udało im się zrealizować. Szkoda, że czas mijał tak szybko i nie było mi dane porozmawiać dłużej z większą ilością scenowców.

Z powodu dość intensywnej soboty, niedzielny poranek nie należał do najłatwiejszych. Na szczęście po posileniu się w hotelowej stołówce przybyło nam nieco energii i całą grupą ruszyliśmy po raz ostatni na zamek. Tego dnia (ciekawe dlaczego) w zamkowej knajpie schodzili już prawie wyłącznie chłodne soki owocowe, zimne wody mineralne i energetyczne, wysokokaloryczne dania, a raczej to, co w



TIMMY SPECY.PL



Screeny z dema „Timmy” grupy specy.pl (demo compo, pierwsze miejsce)

kuchni pozostało, gdyż – z tego, co słyszałem – scenowcy wyjedli i wypili cały dostępny zapas jedła i napitków. Ogołocona została nawet okoliczna „budka” z jedzeniem, która przyjechała specjalnie na party i ustawiła się na dziedzińcu, ale chyba źle oszacowała rozmiary żołądków polskiej demosceny. Na niedzielę w planie imprezy pozostał już tylko jeden, ostatni punkt programu – ceremonia wręczenia nagród i podsumowanie imprezy. Najlepszym demem Riverwash 2014 okazało się „Timmy”, zaś „Sunglasses at Night” z dwoma głosami mniej ulokowało się na miejscu drugim. Dopiero na kolejnych, oddalonych pozycjach znalazło się 5 dem pecetowych. Jak widać, podczas gdy w realnym świecie pecety dominują niepodzielnie, na imprezach demoscenowych stare, kultowe platformy rządzą. W Chip Music Compo nikt nie znalazł sposobu na pokonanie silnej amigowej reprezentacji reprezentowanej przez Acemana i JazzCata. W Streaming Music Compo pierwsze miejsce wywalczył Mercury, z doskonałym, „zadymionym” – nieco jazzowym – bardzo oryginalnym kawałkiem, jakby żywcem wyjętym z czasów prohibicji. W „Twoim starym demo kompo” wygrała grupa Elude, która dotychczas zwana była „grupą drugiego miejsca” (i kto wie czy nie przełamali w końcu swojej złej passy). Pełniejsze wyniki (pierwsze trzy miejsca) znajdują się gdzieś w ramce obok. Warto jeszcze wspomnieć, że nagrodami za premiowane miejsca w każdej konkurencji były bardzo stylowe statuetki. Podczas „ceremonii zamknięcia” wylosowano też szczęśliwca, który otrzymał zwrot kosztów podróży na party, a że okazał się nim – mieszkający w Białymstoku Codi/Ghostown, jego kieszeń zasilona została kwotą adekwatną do niemałej odległości Racibórz – Białystok. Po zakończeniu ceremonii posiedzieliśmy jeszcze chwilę na zewnątrz, dopiliśmy zimne soki i ruszyliśmy w drogę powrotną.

Podsumowując, Riverwash 2014 było imprezą bardzo udaną. Przyznam szczerze, że po wcześniejszych (nie zawsze perfekcyjnych) edycjach tejsze nie spodziewałem się, że tym razem będzie aż tak dobrze. Kto wie, czy nie był to nawet jeden z najlepszych scenowych eventów ostatnich lat w kraju nad Wisłą. Organizacja, miejscówka, atmosfera, ludzie, a nawet pogoda, zagrały koncertowo. Brak rozpraszających koncertów i nachalnych reklam sponsorów, również wyszły tylko na dobre. Pa-



Stylowa nagroda z party, zdjęcie: Slayer



nowie organizatorzy, w składzie: Aceman, Argasek, Borys, Fei, Gorzyga, SuperNoise, Ubik i V0yager – odwalili naprawdę zaciąg robotę, za którą (i za świetnie spędzony czas) dziękuję.

Jak dowiedziałem się (na razie wciąż jeszcze nieoficjalnie) od Argaska pozytywny odbiór party oznacza, że przyszłoroczna edycja *Riverwash* jest wielce prawdopodobna. Niestety ani miejsce, ani data nie zostały jeszcze potwierdzone. Terminowo pod uwagę brany jest początek września, zaś lokalizacyjnie w grę wchodzi albo Kraków albo znów Racibórz. Szczypty pikanterii dodaje fakt, że niedługo po *Riverwash*, restauracja, w której odbyło się party, została na jakiś czas zamknięta, a Za-

mek Piastowski szuka chętnych do jej prowadzenia. Nie ma to jednak nic wspólnego z wybrakami scenowców, gdyż nie dość, że niczego wówczas nie zniszczono, to jeszcze udekorowano salę, a utarg w tych dniach, który „zapewnili scenowcy” był ponoć rekordowy.

W przyszłym roku odbyć ma się także *Retrokomp* z demoscenową częścią *Loadererror*. Czy jest w Polsce miejsce na dwie scenowe imprezy? Wydaje mi się, że tak. *Riverwash* jest z założenia typowo multiplatformowy, z mocniejszym naciskiem na „aspekt towarzyski” spotkań, *Loadererror* mógłby natomiast stać się imprezą znacznie bardziej nakierowaną na Amigę. Trochę to dziwne, że w tak dużym kraju jak Polska, od kilkunastu lat nie udaje się zorgani-

zować amigowego demo party z prawdziwego zdarzenia, a jak pokazuje przykład *Silly Venture*, w światku Atari jest to możliwe. Bez wątplenia zarówno *Riverwash* jak i *Retrokomp/Loadererror* warto będzie w roku 2015 odwiedzić, bo mogą się one dobrze uzupełnić.

Na koniec chciałem jeszcze pozdrowić kolegów z forum PPA, którzy (o ile się nie mylę) odwiedzili scenowe party po raz pierwszy (Flops, Arti i Alt) i z tego co mówili, bardzo im się podobało.

P.S. Gorzyga & Argasek – dziękuję za udostępnienie zdjęć.

Tomasz Pacyna (Slayer/Ghostown)
info@pixeldreams.pl

Wybrane wyniki z party

Chip Music Compo

1. „Go Back To The River!” – AceMan (173 pkt.)
2. „Boobies Everywhere” – Jazzcat/Ghostown (147 pkt.)
3. „Space Dive” – Malfunctor/Altair (125 pkt.)

Streaming Music Compo

1. „This is no time for prohibition” – Mercury (142 pkt.)
2. „Low Frequency Flow” – Dakota/Futuris/Altair (139 pkt.)
3. „Petrolhead” – Jazzcat/Ghostown (134 pkt.)

Oldschool GFX Compo

1. „Watchmaker of Light” – Slayer/Ghostown (171 pkt.)
2. „One Zak and his Kracken” – Carrion (154 pkt.)
3. „ubiquitous” – Jok + PikkuMyy (106 pkt.)

Newschool GFX Compo

1. „The October Country” – Slayer/Ghostown (146 pkt.)
2. „Battle for planet earth” – DKL/Ghostown (131 pkt.)
3. „The Elements” – Biter/Futuris (114 pkt.)

Demo Compo

1. „Timmy” – specy.pl (145 pkt.)
2. „Sunglasses at Night” – Ghostown + Whelpz (143 pkt.)
3. „Burp” – Futuris (116 pkt.)

Produkcje z party można pobrać ze strony <http://demozoo.org/parties/2171/>



Lewo: pierwsze miejsce w Newschool GFX
Góra: drugie miejsce w Oldschool GFX
Lewo-dół: pierwsze miejsce w Oldschool GFX
Dół: drugie miejsce w Newschool GFX



PRODUKT TWÓRCÓW SAKURA 4MB PCMCIA SRAM
JAROSŁAWA BIELIŃSKIEGO I RADOŚŁAWA KUJAWY

DECELERATOR 4030

BEZPIECZEŃSTWO I WYGODA W TWOJEJ AMIDZE
JUŻ W 2015!



EVEN IF OUR PRODUCTS ARE TARGETTING RETRO COMPUTERS, WE BELIVE THAT THEY SHOULD BE BUILT USING MOST MODERN TECHNOLOGIES AVAILABLE (HIGH PERFORMANCE, LOW POWER, ENVIRONMENTALLY FRIENDLY COMPONENTS, LEAD FREE SOLDER, ETC.). TO SUPPORT INNOVATION WITHIN THE RETRO COMMUNITY, WE PROVIDE SCHEMATICS, BOARD LAYOUT FILES AND CPLD CORES OF OUR PRODUCTS UNDER OPEN SOURCE LICENSES. THIS MEANS THAT ANYONE CAN BUILD THE PRODUCT THEMSELVES! ALTHOUGH, THAT WOULD PROBABLY BE MORE EXPENSIVE THAN JUST BUYING IT FROM US. ALL SAKURA RETRO PRODUCTS ARE SOLD ONLY THROUGH OUR EXCLUSIVE DEALER - RETROAMI

SAKURA-IT.PL

