

amiga

měsíčník pro uživatele počítačů amiga

20 Kčs - 3/91

SOUND TRACKER





Nový operační systém

Mnozí z nás začínali na Amize s operačním systémem (zde tím budu nazývat kombinaci Kickstartu a Workbench) verze 1.2 a tak jsme nezažili problémy prvních operačních systémů. Postupem času jsme byli otráveni nemožností spouštět najednou víc než jeden program z Workbench a chybějící možnosti editace v CLI. Za nějaký ten čas se objevila verze 1.3 a mnozí z nás jsme začínali mluvit o revoluci. Workbench 1.3 přinesl možnost editovat v CLI (Shell). Přinesl též nové prvky jako pipelining, nový způsob uchování dat na pevném disku a nesmazatelný pevný ramdisk. Mnozí ale díky nedostatku informací rozvířili hladinu tvrzeními o Kickstart 1.3, jako nezbytné součásti Workbench 1.3. Nakonec jsme se všichni dozvěděli, že nový Kickstart má jen jednu jedinou věc navíc a tou je možnost bootování z pevného disku. Sice se objevily programy, které nepracovaly pod novým Kickstartem nebo naopak pod starým, ale jak jsme všichni později zjistili, bylo to dáno tím, že programátoři těchto programů obcházeli operační systém. Tolik historie a teď bych Vás chtěl zavést to tajů nejnovějšího operačního systému 2.0, který se poprvé objevil ve spojitosti s Amiga 3000. Zde naznačím jen základní rysy, protože o tom chystáme samostatný článek.

Než Vám něco řeknu o novém operačním systému, chtěl bych Vás seznámit s novými čipy, které se ve spojitosti s ním objevily. Většina z Vás už se určitě seznámila s Fat Agnus, který rozšiřuje CHIP RAM na 1 MB. Nové čipy nesou označení ECS (Enhanced Chip Set) a obsahují následující čipy: Super Agnus a Super Denise. Super Agnus rozšiřuje CHIP RAM jen v Amiga 3000 na 2 MB, ale ve

všech Amigách rozšiřuje blitter operace na matici 32Kx32K. Super Denise má čtyři nové módy zobrazování: Productivity (640x480), Interlaced Productivity (640x960), SuperHi-res (1280x200) a Interlaced SuperHi-res (1280x400). Všechny nové módy mají omezenou paletu barev. Maximálně podporují čtyři barvy (dva bitplanes). Ale co je asi nejzajímavější, je to, že Super Denise má programovatelné sync rates (synchronizační hodnoty) a tím mohou programátoři vytvořit své vlastní rozlišení, které budete moci používat na příslušných multisync monitorech. Super Denise má také přepínání mezi NTSC a PAL obrazy, co zvýší výměnu programů mezi Evropou a Amerikou.

Na závěr Vám naznačím, co nového přináší samotný operační systém. Na první pohled všechny uživatele zaujme nový 3D vzhled Workbench a řekl bych, že se už konečně dostal vzhledově na úroveň grafických uživatelských rozhraní jako je např. Open Look. Programátory nejvíce zaujme nová knihovna ASL, která obsahuje standardní funkce pro file a font requester, aby se už konečně všechny aplikace chovaly stejně. Celý Workbench je řešen asynchroně a můžete tedy spouštět programy libovolně a dokonce nemusíte čekat na vykonání zvoleného programu, ale můžete jít dělat hned něco jiného nebo dokonce zastavit činnost programu dříve, než skončí sám. Něco, co zaujme všechny z nás, kteří rádi šetříme místem, je to, že soubory už nemusí mít ikony, ale mohou být spouštěny kliknutím přímo na jméno souboru v Workbench. Nový operační systém obsahuje dlouho očekávaný programovací/manipulační jazyk ARexx, ale o něm až v samostatném článku. Osobně jsem byl zklamán z toho, že tato verze operačního systému ještě neobsahuje vektorové fonty, ale Commodore má v plánu upgrade operačního systému 2.0, který už by je měl obsahovat. Do dnešního dne se nám nepodařilo vypátrat, co bude s uživateli starých Amiga počítačů, jako je A1000 nebo A500 a dokonce A2000. Nový Kickstart 2.0, který je tentokrát nezbytný, má 512K ve dvou čípech. Nastává otázka, co zamýšlí Commodore s těmito uživateli. Už dnes je ale jasné, že tito uživatelé nebudou moci využívat 2MB CHIP RAM pomocí nového Super Agnus. Držme pohromadě a žádejme od firmy širší podporu. Přece nemáme peníze na to, abychom si každých pár let koupili nový počítač.

S pozdravem

Khaled Hussein
šéfredaktor

IMPRESSUM

(c) 1991 "amiga - měsíčník pro uživatele počítačů amiga." Nevyžádané příspěvky se nevracejí.

Ročník II. Cena 20 Kčs. Reg. č. MK ČR 5281. MIČ 46067.

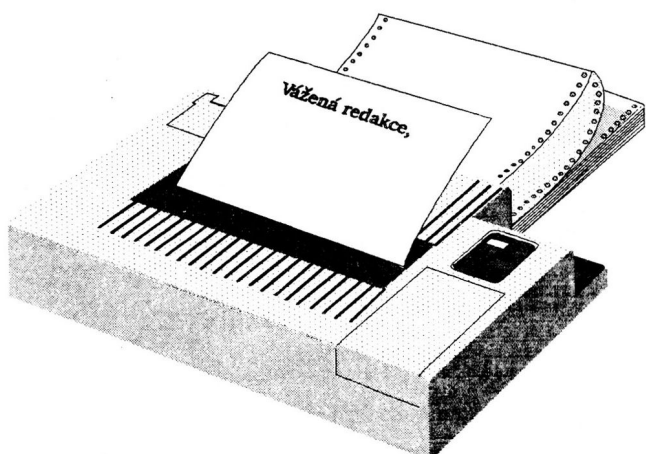
Vydavatel: Jiří Prožr

Šéfredaktor: Khaled Hussein Odpovědný redaktor: Tomáš Stibor

Fotografie: Petr Tůma Layout: Ing. Tomáš Daniček

Tisk: Ústav pro výzkum motorových vozidel

Redakce: Italská 29, 120 00 Praha 2 Inzerce, informace a předplatné: Italská 29, 120 00 Praha 2



Odpovědi na vaše dotazy

Čtenář L.S. z Brna (neuvedl svoji adresu) se nás dotazuje na význam chybového hlášení "Out of heap space". Toto hlášení signalizuje nedostatek místa v pracovní paměti Basicu. Stává se to při použití více oken nebo obrazovek nebo použitím mnoha zahnízděných smyček (např. FOR...NEXT). Náprava se provede tak, že přiřadíme v programu příkazem CLEAR větší pracovní paměť pro Basic.

Basicové programy, které uveřejňujeme v našem časopisu, jsou tvořeny tak, aby byly dobře čitelné i na monochromatických monitorech. Nemusíte je tedy upravovat. U všech těchto programů se předpokládá, že máte v Preferences přepnuto pole TEXT na 80 znaků na řádek.

Programy jsou odzkoušené, ale bohužel, v průběhu tisku se může vloudit chybička. Snažíme se, aby těchto tiskových chyb bylo co nejméně. Na všechny vaše dopisy a dotazy k programům samozřejmě odpovíme. Nezapomeňte uvést svoji adresu včetně PSČ.

Vzhledem k tomu, že při přepisování programů do počítače může dojít k chybám (a jak vyplývá z vašich dopisů, je náš předpoklad oprávněný), rozhodli jsme se, že vám ušetříme práci s přepisováním programů a vydáme na disketě všechny uveřejněné i připravované programy včetně textů. Z velké části zde budou programy z elektrotechniky, ale nezapomeneme i na ostatní obory. Přivítáme vaše náměty k této iniciativě. (bb)

Dotaz: Jak nahrát obrázky vytvořené v Deluxe Paintu do svého programu v AmigaBasicu?

Odpověď: Obrázky vytvořené v kreslicím programu Deluxe Paint jsou v tzv. tvaru ILBM vzájemného přenosu IFF (viz amiga 1/91). Jak tyto obrázky používat ve vlastních programech napsaných v AmigaBasicu je ukázáno v několika ukázkových programech dodávaných na disketě Extras (např. LoadILBM-SaveACBM). Prostudováním těchto příkladů se dozvíte mnoho dalších informací o ILBM respektive ACBM formátu (ACBM-formát, který se používá v AmigaBasic programech).

TEXTOMAT má chybu!

Při práci s tímto textovým editorem jsme objevili dvě malé chybičky, které naštěstí nemají destruktivní vliv na činnost tohoto oblíbeného programu.

Vyzkoušejte si sami. Napište nějaký nadpis a pak jej umístěte doprostřed stránky zvoláním povelu Zentrieren v menu Ausrichtung hlavního menu Einstellung. Nyní ukončete řádek stisknutím klávesy RETURN. Další řádek chcete mít prázdný, t.j. stiskněte opět RETURN a teď se to stane. Vycentrovaný nadpis se posune vpravo. Náprava je jednoduchá. Umístěte textový kurzor do řádku s nadpisem a stiskněte klávesu Help. Řádek bude znovu a správně vycentrován. Potom musíte opustit řádek kurzorovou klávesou (ne s RETURN, jinak se chyba opakuje) a nadpis zůstane neporušen.

Druhá chyba se vyskytuje při psaní úzkým písmem, když máte nastaven počet písmen na řádek větší, než je možné umístit na obrazovce. TEXTOMAT tuto úlohu řeší tak, že posunuje text horizontálně přes obrazovku do prava. Jakmile skočíte z pravého okraje textu na levý okraj (klávesami Alt + levá kurzorová šipka), zmizí obsah tohoto řádku. Jen žádnou paniku. Obsah řádku zmizel jen zdánlivě. Stisknutím klávesy Help se text opět objeví na správném místě na obrazovce.

Nakonec ještě jednu dobrou radu. TEXTOMAT umí psát s odstupem řádků jeden, jeden a půl a dva řádky. O tom, jak velký prostor je vymezen pro řádek rozhoduje nastavení ve formuláři (parametr Zeilenabstand). Pro různé tiskárny se nastavují různé parametry. Tak např. posuv papíru v kroku 1/72 palce mají tiskárny IBM a pak se zde nastavuje hodnota 12, zatímco tiskárny Epson a kompatibilní pracují s posuvem papíru v kroku 1/60 palce a pak je třeba zapsat do formuláře hodnotu 10. Tímto je zaručeno, že program bude správně vypočítávat délku stránky. Podívejte se do manuálu k vaší tiskárně s jakým krokem pracuje a pak запиšte do formuláře správnou hodnotu parametru Zeilenabstand.

Jak vypnout myš?

Další trik, jak vypnout myš v basicovém programu nabízí Amiga-Magazín SRN.

LIBRARY "graphics.library"

CALL FreeSprite(0)

Jakmile pohnete s myší, objeví se opět šipka myši na obrazovce. Pozor na to, aby program měl přístup k volané knihovně.

(Volně zpracováno podle Amiga M&T).

SOUNDTRACKER

RECENZE

SoundTracker je hudobný program fungujúci ako sampler a sequencer. Pôvodným zámerom autorov nebolo vytvorenie programu pre potreby amatérov a poloprofesionálov, ale editora hudby pre hry a intrá. Program pracuje so štyrmi generátormi zvuku a produkuje kvalitnú stereo-produkciu s indikáciou zvukového spektra. Existuje viac verzií (najnovšie dokonca verzia so zmenou názvu - NOISETRACKER), s ktorých novšie majú celý rad vylepšení. Nasledujúci popis sa venuje základným funkciám, ktoré sú vo všetkých verziách zhodné.

Pracovný stôl SOUNDTRACKERu

Po spustení programu sa objaví pracovná plocha, ktorá ponúka všetky potrebné funkcie (nepracuje sa s lištovým menu). V dolnej časti vidíme štyri stĺpce, prezentujúce štyri generátory zvuku. Do nich vpisujeme noty a požadované parametre zvuku (parameter zvuku sa dá meniť v ľubovoľnej fáze produkcie - viď. Editácia skladby). V tejto časti vidíme (vľavo v strede obrazovky) aj malé okienko s číslom 00, určujúcim v ktorom patterne sa nachádzame. PATTERN je celá editovacia časť 4 stĺpcov a 64 riadkov (môžete si ho prezrieť kurzorovými šípkami <- a ->). Na obrazovke máme dva druhy kurzorov - klasickú kurzorovú šípku (ovládanú myšou) a rámček v editovacej časti patternu, ktorým môžeme pohybovať kurzorovými šípkami. Práve tento kurzor slúži pri editácii a ukazuje miesto, s ktorým sa pracuje.

Klávesnica.

Klávesy vašej Amigy sú usporiadané tak, že tvoria dve oktávy klavírnej klaviatúry. Orientujte sa podľa klávesy "Q", ktorá je štandardne nastavená na tón C1. Klávesnica sa dá posúvať o oktávu hore a dole pomocou kláves "F1" a "F2".

Editácia.

Editovacím kurzorom pohybujeme kurzorovými šípkami v strede Amigy. Ak chceme, aby počítač našu editáciu akceptoval, musíme myšou zvoliť ponuku Edit v hornej časti obrazovky (vyššie verzie ST dovoľujú pre tento účel použiť klávesu "Space"). Po tejto voľbe by mala mať šípka modrú farbu.

Funkcie kláves.

(Kurzorové klávesy = kurz)

kurz ... vertikálny i horizontálny pohyb editovacieho kurzoru.

Shift + Kurz ... zrýchlený pohyb kurzoru.

Alt + Kurz ... veľmi rýchly pohyb kurzoru.

F1 ... prvá oktáva (štandardne nastavená).

F2 ... druhá oktáva.

F6 ... nastav kurzor na prvú pozíciu patternu.

F7 ... na 16 pozíciu patternu.

F8 ... na 32 pozíciu patternu.

F9 ... na 48 pozíciu patternu.

F10 ... na poslednú (63) pozíciu patternu.

Shift + F4 ... uchovaj v zásobníku obsah aktuálneho stĺpca.

Shift + F5 ... vlož obsah zásobníku do aktuálneho stĺpca.

Alt + F4 ... v zásobníku uchovaj aktuálny pattern.

Alt + F5 ... vlož zásobník do aktuálneho patternu.

Del ... maž notu pod editovacím kurzorom.

Špeciálna komb. ... *Alt + Shift + ľavá a pravá klávesa myši* = opusti ST.

(platí len pre vyššie verzie) :

Pravá Amiga kl. ... hraj aktuálny pattern.

Pravý Alt ... hraj celú skladbu.

Pravý Shift ... record funkcia (editácia v

reálnom čase).

Alt + Kurz vpravo ... ukáž vyšší pattern.

Alt + Kurz vľavo ... ukáž nižší pattern.

Shift + Kurz vpravo ... zvýš Song-pozíciu.

Shift + Kurz vľavo ... zníž Song-pozíciu.

Shift + TAB ... transponuj nastavený zvuk o poltón hore.

Shift + Ctrl ... transponuj nastavený zvuk o poltón dole.

Alt + TAB ... transponuj všetky zvuky v patterne o poltón hore.

Alt + Ctrl ... transponuj všetky zvuky v patterne o poltón dole.

CapsLock ... vyskúšaj! (HAHAHA).

Numerická klávesnica :

0 ... nastav nástroj \$0.

1. rad ... nastav nástroje \$1 - \$4.

2. rad ... \$5 - \$8.

3. rad ... \$9 - \$c.

4. rad ... \$d - \$10.

Vybratie zvukov.

Od verzie V2.4 môžete použiť až 32 samplingov v jednej skladbe, samozrejme miera použitia záleží od ich dĺžky (v Amige s 512KB prakticky nemôžeme použiť všetkých 32 schránok). Všimnite si funkcie ľavej časti obrazovky - každá má k dispozícii dve šípky, hore a dole. V ponuke "Preset" si môžete prezrieť dostupné zvuky tak, že myšou nabehnete na šípku dole a stlačíme ľavé tlačítko myši. Jednotlivé zvuky sa ukazujú v strede obrazovky v abecednom poradí. Pri použití šípky dole sa vraciame k začiatku.

Po podržaní tlačítka myši sa aktivuje repeat funkcia, čím myš šetríme. Vyššie verzie ponúkajú efektívnejší výber samplingov i výber samplingov nezaznamenaných v PresetList (pozri PLST). Ak sme našli vhodný zvuk a chceme si ho vypočúť, musíme ho nahráť do pamäti Amigy. Aktivujeme ponuku UsePset (NoiseTracker toto odbúra - stačí si vybrať zvuk myšou z predloženého zoznamu po aktivovaní funkcie PLST) čím sa zvuk nahrá do schránky 0, funkcie "Sample". Tu bude mať tento zvuk svoje stále miesto a tiež v skladbách sa bude prezentovať poradovým číslom 0. Pri nahrávaní ďalšieho zvuku postupujeme rovnako, no musíme zmeniť zvukovú schránku, pretože nahraný zvuk by premazal zvuk predošlý. Urobíme tak pomocou už spomínaných šípok. Tento postup dodržiujeme pri nahrávaní všetkých zvukov, ktorých od verzie 2.4 môže byť teoreticky až 32.

Editácia skladby.

Jednou z výhod SoundTrackeru je možnosť nasta-

vovať parametre pre každý zvuk zvlášť a sú prípustné zmeny i počas znenia jedného zvuku. Použijeme Edit-mód a číslu v editovacej časti. Editovací riadok (s príkladnými parametrami)-má tvar :

C#1 3 C 2 3

Hodnoty, určujúce parameter efektu

označenom v riadiacom znaku.

Řídící znak

Číslo zvukovej schr., ktorej zvuk znie.

Nota, ktorá zaznie.

Nota : určuje výšku tónu, edituje sa buď priamo, alebo pomocou funkcie "Record"

Číslo zvuku : určuje číslo zvukovej schránky, ktorej zvuk v danej chvíli zaznie. Dosadzuje sa automaticky pri editácii.

Riadiaci znak : určuje jaký efekt sa s aktuálnym zvukom prevedie.

Riadiaci znak môže mať tvar :

0 - normálny stav, prípadne arpeggio.

1 - portamento hore

2 - portamento dole

3 - tónové portamento (platí len pre

4 - vibráto vyššie verzie)

B - skok v patterne

C - nastavenie hlasitosti

D - prerušenie hry v patterne a skok na jeho začiatok

E - nastavenie filtra (E01 = filter nastavený)

F - nastavenie rýchlosti (štandardne 6, 1 = najrýchlejšie)

+ pri uvádzaných príkladoch si všimajte len posledné tri čísla riadiacej časti, pretože tie sú podstatné.

(0) Arpeggio - pekný efekt chvenia tónu. V posledných dvoch znakoch určíme frekvencie striedania kmitočtov.

(1) Portamento - plynulé zvyšovanie, alebo znižovanie frekvencie tónu.

príklad : 1. D#2 3208 - plynule klesá po \$08

2. ----- 0208 - ---- " ----

3. ----- 0208 - ---- " ----

4. ----- 0208 - ---- " ----

1. C#2 31AD - plynule stúpa po \$AD

2. ----- 01AD - ---- " ----

3. ----- 01AD - ---- " ----

4. ----- 01AD - ---- " ----

(B) Skok v patterne : Touto funkciou nám je umožnené prerušiť hru z určeného patternu a ukazovateľ preniesť na iné miesto.

príklad : 1. E-2 3B04 - preruš hru v tomto patterne a

skoč na pozíciu 04.

(C) Nastavenie hlasitosti - po zadaní riadiaceho znaku "B" môžeme nastavovať intenzitu zvuku počas produkcie. Hodnoty môžu byť v rozmedzí \$00(najslabšie) a \$64(najsilnejšie). Štandardne je sila zvuku nastavená na \$64.

príklad : 1. C-1 1C22 - nastavenie hlasitosti na \$22.

(E) Nastavenie filtra : Prikaz umožňuje aktivovať filter.

príklad : 1. A#2 1E01 - filter nastavený

12. G-2 5E01 - filter odstavený

(F) Nastavenie rýchlosti : Umožňuje obmieňať rýchlosť skladby a to i počas produkcie v intervale od \$01(najrýchlejšie) do \$0f(najpomalšie).

Skladanie patternov.

K tejto časti pristúpime vtedy , ak už máme vytvorených viac patternov a chceme ich použiť vo vlastnej skladbe. Amige potrebujeme nejakým spôsobom oznámiť , ktoré patterny a v akom poradí majú zaznieť. Použijeme na to funkcie pracovného stola - Position , Pattern a Lenght , nachádzajúce sa v ľavom hornom rohu obrazovky. Pre nastavenie hodnôt použijeme už spomenuté šípky. Činnosť týchto funkcií si ozrejníme na príklade. Dajme tomu , že chceme aby nám ako prvý zaznel pattern č.3 , potom pattern č.2 a toto sa opakovalo dookola. Použijeme nasledovný postup :

1. Position nastavíme na 0
2. Pattern nastavíme na 3
3. Position nastavíme na 1
4. Pattern na 2
5. Lenght na 3

Position určuje pozíciu , na ktorej sa nachádza čítač a pattern určuje číslo patternu hrajúceho na tejto pozícii. Lenght oznamuje počítaču koniec skladby , teda ak hodnota v Position dosiahne hodnotu určenú v Lenght-1 , skladba sa opakuje.

PLST.

Pýtate sa ako SoundTracker vie aké sampliny môže použiť? Ak sa na disketu pozriete pozornejšie (dir) zistíte , že sa na nej nachádza súbor s názvom "PLST". Práve v tomto súbore je uložený zoznam dostupných samplíngov. Tiež sa na diskete nachádza program (nemusí sa - záleží na disketovej verzii) , pomocou ktorého vieme PLST upravovať (nazvime ho "PRESET-ED"). Existuje viacero verzií , nezameriam sa preto na konkrétny opis , ale prácu v ňom zovšeobecniť.

Po nahratí SoundTrackeru Amiga hľadá súbor PLST. Ak ho nenájde , nižšia verzia SoundTrackeru nebude môcť použiť , pomocou funkcie "USE PSET", zvuk žiaden. U vyšších verzií ST sa zvuk dá nahráť zvlášť funkciou "Load-Sample" , no používanie tohoto spôsobu je oveľa zdĺhavejšie. Predpokladajme , že sa programovaním hudby chcete zaoberať serióznejšie. Určite si vytvoríte datové diskety , na ktorých budete mať len sampliny (poznám ľudí , čo ich majú vyše 30 - iste! je to dosť , no

uvedomte si , že kvalitne digitalizovaný zvuk požiera pamať veľmi intenzívne). Datovej diskete pre SoundTracker VŽDY musíme dať názov v tvare ST-číslo od 01 - 99 napríklad ST-01,ST-12... Zdrojová disketa SoundTrackeru má názov ST-00 a aj na ňu je možné niekoľko samplíngov nahráť. Teraz niečo o štruktúre samplíngov. Na digitalizáciu zvuku používame zvláštne programy (SoundSampler,68000 Digitizer ...) , ktoré za pomoci zvukového digitizéra umožňujú previesť zvukový , analógový signál do číselnej formy a ďalej s ním pracovať v tejto podobe.

SoundTracker potrebuje okrem hlavných dát poznať i dĺžku samplíngu - LENGTH , hodnotu repeat funkcie - REPEAT a mnou zatiaľ neobjasnená funkcia - REPLEN , ktorá sa štandardne nastavuje na hodnotu 2. REPEAT sa používa pri nekonečných zvukoch a určuje , odkiaľ sa má samplíng po doznení opakovať. Hodnoty REPEAT a LENGTH určí digitizér , a keď sme spokojný s vytvoreným (prípadne modifikovaným) zvukom , tieto hodnoty si odpíšeme pretože ich použijeme v PRESET-EDítore. Spôsob zápisu samplíngu do PLST ukážem na príklade. Dajme tomu , že chcem zaznamenať samplíng s názvom "BASSDRUM1" s hodnotami LENGTH = 1022 ; REPEAT = 0 (basový bubon nepotrebuje opakovanie , pretože znie len raz) a REPLEN = 2. Najskôr si samplíng nahráme na datovú disketu (dajme tomu že ST-01). Nahráme PRESET-EDitor , do ktorého všetky údaje zapíšeme (názov diskety na ktorej sa zvuk nachádza - ST-01 , názov samplíngu - "BASSDRUM1" , LENGTH , REPEAT a REPLEN). V PLST sa teda nenachádzajú vlastné data samplíngu , ale len ukazovateľ na jeho disketu , názov a ďalšie tri údaje o ktorých sme si hovorili. Na stretnutiach Amigistov sa vyhlasujú súťaže o najobsiahlejší PLST-súbor , ktorý u niekoho pojmá i 3000 samplíngov. Ja mám síce len niečo vyše 700 , no nevzdávam sa. Existujú dokonca programy , umožňujúce "kradnúť" z hier a intier zvuky , i celé skladby a použiť ich v SoundTrackeri. Ale o nich inokedy.

Play-routine.

Možno že na vašej ST-diskete nájdete ASCII súbor s podobným názvom (STplay-routine.S , REPLAY.S , STPlayer.S , Replayer.S). Ak sa naň pozriete , zistíte že je to assemblerovský zdroják , pravdepodobne dodržiavajúci syntax SEKA-Assembleru. Pomocou tohoto programu , si môžete prehrať vašu hudbu , no na disketu ju musíte uložiť príkazom SaveModule. SaveModule nenahráva len data skladby a ukazovatele na diskety s potrebnými samplíngami , ale nahrá všetko dohromady , play-rutina dokáže túto štruktúru u prečítať a podľa nej uloženú skladbu zahrať. Existujú aj kompilované verzie play-rutín , no ich použitie je dosť obmedzené. Výhodné je vlastniť play-rutinu v nekompilovanom tvare , pretože umožňuje použiť nami vytvorenú skladbu vo vlastných programoch (dovoliť si tvrdiť že 90% hudby znejúcej vo vašich intrách a hrách je komponovaná v SoundTrackeri). Prajem vám veľa úspechov , zábavy i poučenia pri komponovaní hudobných diel.

Rado Spišiak

Mezi grafickými programy pro Amigu, zůstává DeluxePaint standardem, s kterým jsou ostatní programy porovnávány. Je to základní nástroj výtvarné palety Amigy; skutečně, mnoho Amiga uživatelů zakoupilo tento počítač právě pro práci s DeluxePaintem. Ale Electronic Arts a programátor Dan Silva neusnuli na vavřínech. DeluxePaint III - tato jeho poslední verze doznala vítaných kosmetických úprav a přidavků některých nových úžasných vlastností, včetně snadné animace. DeluxePaint III má stále všechny rysy, které dělaly dřívější verze tak populární. Mnoho funkcí, jako např. tvorba mnohoúhelníků, spray, práce s textem a vyplňování plochy bylo posíleno. Nové funkce vám umožní vyjmout část obrazu a s ním kreslit či mazat, pokřivit ho, změnit velikost, rotovat, vytvořit obrys či přebarvit, nebo jej navinout na libovolný mnohoúhelník. Nejlepší ze všeho je to, že každá funkce je rychlejší než její předchůdce v DeluxePaint II.

Prostupné (pull-down) menu kontroluje práci i formát zobrazení; volitelné možnosti malby, jako jsou kmitání, rotace a ohýbání, a efekty, jako jsou rozmazání, míchání a změkčení. Speciální rysy zahrnující perspektivní, komplexní, z klávesnice řízený systém pro naklápění malování či obrazu k vytvoření iluze pohybu do hloubky a vytvoření šablon pro maskování vybraných barev nebo ploch tak, aby nemohly být přemalovány. Je také "ušetřena" obrazovka pro palcové nákresy a podporu pro superbitmap obrázky až do velikosti 1008 x 1024 bodů.



DeluxePaint III vytváří podporu pro mód Extra Half-Brite. S Half-Brite můžete užívat dalších 32 barev v lo-res i interlace rozlišení, které jsou z polovice tak jasné než barvy prvotní palety. Příznivci desktop videa rádi uslyší, že DeluxePaint III vám umožní malovat přímo do overscan prostoru obrazovky - prostoru, který je za normálních okolností zobrazen, jako černý rámeček. Fill tool má nové možnosti, jako jsou průhledné vybarvování a ovinování, které umožňují průběžné vymalování a navíjení hodící se na jakýkoliv tvar. DeluxePaint III také nabízí některé skutečné novinky: možnost malovat ve čtvrté dimenzi, čase. Užitím animačních nástrojů DeluxePaintu III můžete tvořit pohybové ukázky, představy a simulace - např. kreslené filmy typu Králíka Rogera. Animace DeluxePaint III užívá proces zvaný page flipping. Jestliže jste se někdy probírali stránkami "pohyblivých obrázků", pak chápete tuto koncepci. Tvoříte serii obrázků užitím standardních kreslicích nástrojů. Když je sekvence hotová, jsou jednotlivé obrazy uloženy do paměti. Pro spuštění animace řeknete DeluxePaintu III aby postupně celou sekvenci zobrazil předem nastavenou frekvencí a to až do třiceti snímků za sekundu. Na rozdíl od většiny animačních programů se s animačními nástroji DeluxePaintu III zachází snadno. Z Anim menu vyberete počet snímků ve vaší animaci.

DPaint III

Potom malujete na každý snímek. Snímky mohou být děleny, kopírovány nebo spojovány do jakékoli sekvence. Užitím funkce Anipainting jsou snímky pomocí myši postupně zobrazovány přičemž probíhá automaticky animační proces. Můžete též užít malých animací, jako štětců tvorbou animace za použití Animbrush a přibrání celé animované oblasti, jako obyčejný štětec. Animbrush může být užít podobně jako normální štětec, ale s některými nápadnými výsledky. Jedním z dalších příjemných rysů DeluxePaint III je to, že vykonává mnoho obtížné práce za vás. To zahrnuje plynulý pohyb objektu na pozadí; zoomování podél osy z; a provádění komplikovaných rotací, letů a pohybu po dráze v jakékoli kombinaci tří prostorových rozměrů. Klíčem k těmto trikům je Move box, který integruje "custombrush", perspektivu a animaci v jeden snadno ovladatelný celek. Tyto možnosti otvírají možnosti k tomu, aby byl DeluxePaint III užít k tvorbě animovaných titulků pro desktop video. Když jste vytvořili svou mistrovskou animovanou sekvenci, můžete ji spouštět vpřed, vzad nebo ve smyčce. Animace jsou ukládány ve formátu Aegis-Sparta ANIM, který může být čten většinou Amiga animačních programů. Můžete nahrávat své DeluxePaintIII animace do jiných programů nebo si hrát s nimi pomocí hracích utilit dodávaných firmou Electronic Arts. Je však stále několik věcí, které bych rád v DeluxePaint III viděl. Je to například podpora módu hold-and-modify (HAM), automatický scrolling obrazovek, superbitmap a mnohočetná okna s proměnnou velikostí. Přesto zůstává DeluxePaint III nejlepším grafickým programem pro počítač Amiga. Nyní s animačními doplňky je také nejlepším animačním programem. Program vyžaduje, abychom-li provádět animaci 1MB RAM.

Tomáš Macourek
(volně přeloženo podle
časopisu COMPUTE.)

PRO ZAČÁTEČNÍKY

PREFERENCES

Tomáš Stibor

Dnešní sezení u počítače bude sledovat co se v něm děje po zapnutí síťového vypínače. Než dáte do stroje disketu je na obrazovce nakreslena ruka držící disk Workbench. Tento obrázek je uložen v paměti ROM v počítači.

Paměť ROM si můžeme představit jako krabičky, z nichž každá má osm přihrádek a na boku pořadové číslo - adresu, aby se v nich počítač vyznal. V každé přihrádce může být uložena jednička nebo nula. To v osmi přihrádkách dělá celkem 256 možných kombinací nul a jedniček. Krabice má průhledné víko, takže se do ní může počítač podívat, ale nejde v ní nic změnit. Obsah je dán již při výrobě a zůstává v paměti i bez napájení z elektrického zdroje. Paměti ROM v počítači není mnoho a proto obsahuje jen jednodušší programy.

Z diskové mechaniky se ozývá jak počítač čeká na disketu. Po jejím vložení si přečte tzv. zaváděcí blok (boot block). To je na disketě přesně určené místo. Pokud na něm počítač nenajde nic, nebo něco, čemu nerozumí, (zatím toho moc neumí), tak si neví rady a vrátí se k zobrazení obrázku s rukou. V opačném případě udělá to, co si přečte z diskety. Dejme tomu, že to byla normální disketa bez úvodních efektů. Počítač si tedy najde direktorář devs a v něm soubor system-configuration. V něm je uloženo počáteční nastavení. Tento soubor je vytvářen programem Preferences, o kterém si teď pohovoříme.

Po spuštění Preferences - cvaknutím na jejich ikonu - se vám nakreslí výběrové okno s možností další volby (u verze 1.3) nebo se dostanete rovnou na nastavovací tabuli (verze 1.2). U V1.3 spusťte tu část Preferences, kde potřebujete dělat nějaké změny, ale stejně se nakonec dostanete na základní nabídku.

Nabídka zleva:

- Reset All: nastavení všech položek podle výrobce
- Last Saved: pokud se vám něco nepovede, můžete začít znova od nastavení, jak je nahráno na disketě
- Nastavení současného datumu. Nemáte-li rozšíření paměti na 1Mb nebo více s hodinami, tak zde bude nastaven čas a datum posledního záznamu nastavení Preferencí plus čas od zapnutí Amigy. Ale i když interní hodiny zálohované baterií máte, tak se nelekejte, nepůjdou-li vám přesně, ale jen v závislosti na dobré vůli Amigy. Počítači se musí věřit jen do určité míry.
- Políčko Change serial je pouze pro ty, kteří používají tiskárnu nebo jiné zařízení (např. modem) připojené na sériový port počítače. Dostanete se s ním do další části programu. Viz níže.
- Text 60 nebo 80 písmen na řádek. Větší písmo pro ty, kteří používají televizi místo monitoru.
- Nastavení barev. Všechny barvy, které existují, je možno složit ze tří základních barev - modré, zelené a červené. Anglicky - blue, green, red. Z toho vznikla zkratka RGB. Amiga může nastavit kteroukoliv z těchto barev v 16ti odstínech. Celkem je to 4096 barev.

- **Reset Colors:** Návrat na původní barvy před začátkem experimentování.
- **Key Repeat Delay:** Doba, po které se začne při stisknutí tlačítka jeho funkce opakovat. Když chcete například podtrhnout text v dopisu, tak nemusíte stále mačkat tlačítko, ale jen držet. Short > Krátký Long > Dlouhý
- **Key Repeat Speed:** Jakou rychlostí se bude dít opakování. Slow > Pomalu, Fast > Rychle
- **Nastavení umístění obrazu na stínítku monitoru nebo televize.** Posunováním trojúhelníčku si umístíte obraz tak, abyste viděli až do krajů.
- **Workbench Interlace:** Zapíná a vypíná prokládání řádkování. Nemá okamžitý efekt. Projeví se jen tehdy, když nastavení nahrajete na disketu a počítač znovu spustíte.
- **Sloupec s čísly 1, 2, 4** volí poměr mezi dráhou myši po stole a ukazovatelem (pointrem) na obrazovce.
- **Druhý sloupec** nastavuje dobu, po kterou počítač čeká na druhé stažení levého tlačítka myši, aby to bylo signálem k např. otevření diskety.
- **Políčko Change Printer** určuje detailní nastavení pro tiskárnu. Dostanete se s ním do další části programu. Viz níže.
- **Políčko Edit Pointer** slouží ke změně ukazovátka myši. Dostanete se s ním do další části programu. Viz níže.
- **Save:** Nahrát nastavení preferencí na disk a opuštění programu.
- **Use:** Použít nastavení při další práci s počítačem, ale bez jeho uchování. Po vypnutí stroje přestane existovat.
- **Cancel:** Zrušit veškeré experimentování a ukončit program s původním nastavením z diskety.

Tabule pro nastavování sériového portu:

- **Baud Rate:** Rychlost přenosu.
- **Read Bits:** Počet čtených bitů Amigou
- **Stop Bits:** Počet stopbitů.
- **Handshaking:** Doslova "potřásání rukou". Nastavuje se způsob, jakým si spojené stroje navzájem hlásí průběh přenosu dat. To je nutné například k tomu, aby rychlý počítač nezahltl tiskárnu informacemi.
- **Buffer Size:** Velikost paměti dat. Velký buffer zvyšuje rychlost přenosu, ale zabírá místo.
- **Write Bits:** Počet vysílaných bitů.
- **Parity:** Zapnutí a vypnutí vysílání paritního bitu pro kontrolu přenosu.
- **OK:** Zapamatování nastavených údajů a návrat na předchozí tabuli.
- **Cancel:** Návrat bez uskutečnění změn. Další informace o sériovém přenosu, zapojení konektorů a kabelů je v našem časopise č.1/91 str.10 a 11.

Tabule pro nastavení tiskárny:

- **Parallel:** tiskový výstup na paralelní konektor.

- **Serial:** tiskový výstup je směřován na sériový konektor.

Na tabuli vpravo nahoře vyberte ovladač pro vaši tiskárnu. Na disketě Extras jsou další možnosti, v případě Workbench 1.3 prakticky všechny. Ten, který budete potřebovat si překopírujte a v preferencích vyberte. Pozor, ovladače (drivers) se nedají měnit mezi Amigou 1.2 a 1.3. Pokud máte atypickou tiskárnu je možno ovladač vyrobit programem PrtDrvGen. Jestliže si vytvoříte nebo získáte ovladač jiným způsobem, tak zvolte tiskárnu Custom a do okénka Custom Printer Name napište název tohoto ovladače.

- **Paper Size:** Nastavení délky papíru. Čtyři normalizované velikosti a položka Custom. Při ní platí počet řádek na stránku, který si napíšete do položky Length (délka).
- **Left Margin:** Levý okraj. Při vyšším čísle než jedna se na kraji řádků vytiskne nejprve odpovídající počet mezer.
- **Right Margin:** Pravý okraj. Počet znaků na řádce než se přejde na novou.
- **Paper Type:** Druh papíru. Fanfold - skládaný nebo role. Single - jednotlivé listy.
- **Pitch:** Rozteč písma. 10, 12, 15 znaků na palec (2.54 cm)
- **Quality:** Typ písma. Draft - obyčejné, rychlý tisk Letter - pěkné, pomalý tisk
- **Spacing:** Řádkování. Šest nebo osm řádek na palec.
- **Graphic:** Další tabule pro nastavení grafického tisku.
- **Graphic 2 (Jen verze 1.3)** Další možnosti nastavení grafiky tisku.
- **OK:** Zapamatování nastavených údajů a návrat na předchozí tabuli.
- **Cancel:** Návrat bez uskutečnění změn.

Nutno ovšem předeslat, že ne všechny tiskárny umožňují veškeré nabízené možnosti. Pokud máte opravdu velice nestandardní přístroj, použijte ovladač "generic", který by měl umožnit alespoň základní tisk písma. S grafickým tiskem obrázků je to ještě horší. Tisk je pomalý a ne vždy přesný. Jsou však programy (např. FinePrint), které preference obcházejí a tisknou lépe. Pro málo rozšířené tiskárny jsou však nepoužitelné. Pochopitelně, tiskárny s typovým kolečkem také takto obrázky netisknou.

Tabule grafického tisku:

- **Threshold:** Nastavení rozhodovací úrovně. Která barva bude při černobílém tisku považována ještě za bílou a která již za černou. Light - Světlý, Dark - Tmavý.
- **Aspect Horizontal:** tisk vodorovně na papír Vertical tisk na výšku. Obrázek může být větší
- **Shade**
 - Blac and White:** Černobílý tisk
 - Gray Scale:** Převod barev na obrazovce do šedé stupnice
 - Gray Scale2:** Totéž s jiným poměrem převodu (jen u V1.3)
 - Color:** Barevný tisk. Pokud to umí.

Image Positive: Obrázek bude tak, jak ho vidíte.
Negative: Obrázek bude negativní, tedy bílá na obrazovce bude černá na tiskárně a naopak.

OK: Zapamatování nastavených údajů a návrat na předchozí tabuli.

Cancel Návrat bez uskutečnění změn.

Nastavovací tabule Graphic 2 je použita jen v preferencích na disketě Workbench 1.3 a zabývá se podrobným převodem barev, rozměry a umístěním obrázku, které asi málokdo využije.

Tabule ukazovátka myši - Pointer:

Opět zde můžete nastavit další tři barvy, ze který se může pointer skládat. Čtvrtá barva je podkladová.

- Restore: Obnovení původního tvaru pointeru.
- Clear: smazání kreslicí plochy.
- Reset Color: Návrat na počáteční nastavení barev.
- Set Point: Označení místa, které bude na tvaru ukazovátka mít rozhodovací účinky. Tento bod musíte potom vždy umístit tak, aby byl např. uprostřed ikony, kterou chcete otevřít.
- OK: Zapamatování nastavených údajů a návrat na předchozí tabuli.
- Cancel: Návrat bez uskutečnění změn.

Tímto způsobem máte tedy možnost nastavit si na jednotlivých disketách barvy pracovní plochy a další detaily. Pokud budete chtít změnit něco na disketě, na které není program Preferences umístěn, počínejte si následovně.

Je-li na disketě dost místa, Preferences překopírujte z Workbench, použijte a opět smažte. Není-li k dispozici místo, tak nejprve spusťte Amigu z diskety, kterou chcete změnit. Následně do diskové mechaniky vložte disketu Workbench. Tu otevřete a spusťte program Preferences. Disketu opět vyměňte a použijte Preferences jak je uvedeno výše.

V případě, že disketa vůbec Workbench okno nepoužívá, je nutné spustit preference z okna CLI. Máte-li jen jednu diskovou mechaniku, začněte opět disketou, která má být změněna. Po otevření okna CLI (může se jmenovat jinak), napište: df0:prefs/preferences, a opět postupujte stejně. Ještě může nastat případ, že na disketě není ani CLI (např. u různých balíků utilit) a nastavení barev a počtu znaků, popř. tiskárny vám nevyhovuje. V tom případě je asi nejjednodušší vzít kopii Workbench, na ní si nastavit potřebné parametry a na konkrétní disketu překopírovat jen soubor system-configuration. Nezapomeňte, že při změně tiskárny se musíte přesvědčit, zda je na cílové disketě i odpovídající ovladač a v opačném případě ho též překopírovat. Jinak je zbytečné mít na disketě ovladačů několik. Málokdo má více tiskáren, které často střídá a soubory zabírají místo. Doufám, že nyní se vám podaří nastavit preference na vašich disketách ke vší spokojenosti. ☐

JAZYK C

PROGRAMOVACÍ JAZYK C

část 5.

Ing. Tomáš Daniček

V posledním pokračování našeho seriálu o programovacím jazyku C jsem se zmínil o důležitých knihovních funkcích printf a scanf. Uvažujme nyní o skupině programů, které budou vykonávat jednoduché operace se znaky. Záhy zjistíme, že mnohé z rozsáhlejších programů jsou jen rozšířením těchto základních programů.

Vstup a výstup znaku

Standardní knihovna obsahuje funkce pro čtení a vypsání znaku. Funkce getchar načítá vždy další znak pokaždé, když je vyvolána a jako hodnotu vrací tento znak. Takže po příkazu

```
c = getchar()
```

obsahuje proměnná c další znak ze vstupu. Znaky jsou obvykle zadávány z klávesnice, ale to nás nyní nemusí zajímat. Funkce putchar je doplňkem funkce getchar

```
putchar(c)
```

Tato funkce vytiskne obsah proměnné c na nějaké médium - obvykle na obrazovku. Volání funkce putchar a printf můžeme kombinovat. Stejně jako v případě funkcí printf a scanf není na funkcích getchar a putchar nic magického. Nejsou součástí jazyka C, ale jsou z něho dosažitelné.

Kopírování souboru

Leckterému uživateli AMIGY se při pojmu kopírování vybaví často jediný způsob získávání kvalitního softu od svých známých. Ale já mám na mysli poněkud jednodušší problém.

Znáte-li funkce getchar a putchar, můžete napsat mnoho užitečných programů, aniž budete vědět něco dalšího o operacích vstupu a výstupu. Nejjednodušším příkladem je program, který kopíruje vstup do výstupu po jednom znaku.

Vývojové schéma vypadá takto:

```
PŘEČTI ZNAK
WHILE (ZNAK NENÍ SYMBOL PRO KONEC
SOUBORU)
  VYPIŠ ZNAK
  PŘEČTI DALŠÍ ZNAK
```

Program v jazyku C bude vypadat takto:


```
main() /* Kopírování vstupu na výstup: 1. verze */
{ int c;
  c = getchar();
  while (c != EOF) {
    putchar(c);
    c = getchar();
  }
```

Operátor != znamená "NEROVNÁ SE".

Hlavním problémem je zjistit, byl-li načten konec vstupu. Obvykle je dáno konvencí, že když funkce getchar narazí na konec vstupu, tak vrací hodnotu, která není normálním platným znakem. Jediný, ale zanedbatelný problém je to, že existují dvě konvence pro indikaci konce souboru. My jsme se této nepříjemnosti vyhnuli tím, že používáme symbolické jméno EOF pro hodnotu "KONEC SOUBORU", ať už je jakákoliv. V praxi se EOF buď rovná -1 nebo 0, a proto musí být na začátku programu definice

```
#define EOF -1
```

nebo

```
#define EOF 0
```

(Tato definice bývá obvykle v hlavičkovém souboru stdio.h) Použitím symbolického jména EOF, které reprezentuje hodnotu funkce getchar pro načtení konce vstupu, jsme zajistili, že jen jediná řádka v programu závisí na konkrétní číselné hodnotě.

Současně musíme deklarovat proměnnou c typu int, ne c h a r, aby mohla obsahovat hodnotu jakou funkce getchar vrací. Jak dále uvidíme, tato funkce je skutečně typu int, protože musí být schopna vracet nejenom znaky, ale také reprezentaci symbolu EOF.

Program pro kopírování může být zkušenějšími programátory v jazyku C napsán stručněji. V tomto jazyku může být každý příkaz, jako např.

```
c = getchar()
```

použit ve výrazu, jehož hodnota je prostě rovna hodnotě, která je přiřazována levé straně výrazu. Jestliže je přiřazení znaku proměnné c vloženo na pozici podmínky v příkazu while, tak program pro kopírování souborů může být napsán takto:

```
main() /* Kopírování vstupu na výstup 2. verze */
{ int c;
  while((c = getchar()) != EOF)
    putchar(c);
}
```

Program přečte znak, přiřadí ho proměnné c a testuje, zda tento znak byl příznakem konce souboru. Jestliže nebyl, tak tělo cyklu while je vykonáno. Testování se znovu opakuje. Když se narazí na konec souboru, příkaz while, stejně jako funkce main ukončí činnost.

Tato verze programu soustřeďuje vstup na jedno místo - nyní už je jen jedno volání funkce getchar. - a zkracuje text programu. Vložení přiřazovacího příkazu do testovacího výrazu je jednou z možností, kdy jazyk C umožňuje významné zkrácení textu programu. (Je možné se o tuto možnost nestarat a vytvářet "nevnořující se" text programu, ale tomu se budeme snažit vyhýbat.)

Je důležité si uvědomit, že vložení přiřazovacího příkazu do závorek je opravdu nezbytné. Priorita operátoru != je vyšší než přiřazení =, z čehož vyplývá, že při

nepřítomnosti závorek bude příkaz != vykonán dříve než přiřazení =. Proto příkaz

```
c = getchar() != EOF
```

je shodný s příkazem

```
c = (getchar() != EOF)
```

To má za následek to, že proměnné c bude přiřazena hodnota 0 nebo 1 podle toho, zda funkce getchar narazila na konec souboru nebo ne.

Počítání znaků

Následující program počítá znaky. Je to úprava programu pro kopírování.

```
main() /* Počítání znaků na vstupu */
{ long nc = 0;
  while (getchar() != EOF)
    ++nc;
  printf("%ld\n", nc);
}
```

Příkazem ++nc uvádíme nový operátor, ++, který provádí zvětšení o jedničku. Můžeme ovšem rovněž napsat nc = nc + 1, ale ++nc je stručnější a mnohem efektivnější z výpočetního hlediska. -- je obdobný operátor pro odečítání jedničky. Operátory ++ a -- se mohou objevit buď před proměnnou (++nc) nebo za ní (nc++). Tyto dva tvary mají ve výrazech různý význam, jak uvidíte v dalších pokračováních, ale jak ++nc tak i nc++ zvětšuje proměnnou nc o jedničku. V našem příkladě budeme psát tyto operátory před proměnnými.

Program pro počítání znaků sčítá znaky v proměnné nc, která je typu long int /proměnná s dvojnásobným počtem bytů/. Chceme-li počítat s mnohem většími čísly, můžeme použít proměnné typu double (proměnných s pohyblivou čárkou, které mají proti proměnným typu float dvojnásobnou délku). Také použijeme příkaz for místo příkazu while, abychom ukázali jiný způsob tvorby cyklu.

```
main() /* Počítání znaků na vstupu */
{ double nc;
  for(nc = 0; getchar() != EOF; ++nc)
    ;
  printf("%.0f\n", nc);
}
```

Specifikace %f ve funkci printf je používána jak pro proměnné typu float, tak i typu double. %.0f potlačuje tisk neexistující desetinné části čísla nc.

V tomto programu je tělo cyklu for prázdné, protože veškerá činnost je soustředěna na testování a reinitializaci. Ale gramatická pravidla jazyka C vyžadují, aby tělo cyklu for existovalo. Izolovaný středník zde představuje prázdný příkaz a tak je pravidlo splněno. V programu jej píšeme na samostatnou řádku, abychom si ho snadněji všimli. Předtím, než program pro počítání znaků opustíme a ukončíme dnešní pokračování našeho seriálu, všimněme si, že když vstup neobsahuje žádné znaky, tak je podmínka cyklu while nebo for nepravdivá hned při prvním vyhodnocení a proměnná nc je rovna nule, což je správný výsledek. Jednou z příjemných vlastností příkazu while a for je to, že podmínka je testována na začátku cyklu, předtím, než je vykonáno tělo cyklu. □

ILBM v praxi

Khaled Hussein

V článku o standardu vzájemného přenosu IFF, který vyšel v Amiga 1/91, jste si mohli přečíst o formátu dat ze strany teorie. V druhém pokračování, které bylo uveřejněno v Amiga 2/91, jste se mohli seznámit s některými důležitými systémovými strukturami, které jsou nezbytné pro zpracování obrázků. V tomto posledním díle bych chtěl všechny tyto informace shrnout to jednoduchého programu, který čte obrázky ve formátu ILBM.

Program není vůbec optimalizován a skýtá hodně prostoru pro vylepšení. Záměrně pracuje jen s obrázky v rozlišení 320x256 v 32 barvách (pět bitplanes). Jako cvičení Vám ponechávám rozšíření programu na všechny grafické módy a na využívání speciálních efektů, jako je *color-cycling* (pseudoanimace prováděná výměnou barev v paletě barev).

Teď bych Vás rád provedl programem, abyste získali alespoň základní představu o tom, jak pracuje, a tím měli snadnější práci při jeho vylepšování. Tak za prvé je celý program rozdělen do čtyř souborů: *inclall.h*, *iffypes.h*, *readIFF.c* a *imaging.c*. První ze souborů (*inclall.h*) jen vkládá potřebné include soubory pro práci s knihovnami *intuition* a *graphics* a některé základní systémové struktury. Druhý ze souborů (*iffypes.h*) je include soubor, který obsahuje nezbytné typy, konstanty a prototypy funkcí. Na řádkách 10 a 11 definujeme dvě konstanty, které budeme používat jako hodnoty, které budou funkce vracet podle toho, jak úspěšně splnily své úkoly. Na řádce 13 definujeme makro, pomocí kterého budeme definovat jednotlivé konstanty typů (řádky 15 až 19), které jsou nezbytné pro sestavení obrázku. Na řádkách 21 a 22 jsou definovány dvě konstanty komprimace, podle kterých hlavní program zjistí, zda je obrázek komprimován nebo ne. Na řádkách 24 až 29 je definována struktura, která obsahuje základní údaje o obrázku, tj. šířku, výšku, počet bitplanes, komprimace, šířku a výšku stránky. Na zbývajících řádkách jsou prototypy použitých funkcí.

Pro shrnutí: tyto dva soubory jsou tzv. include soubory a nikdy je nekompilujeme. Používáme je jen pro zpřehlednění veškerého dění kolem našeho programu.

Teď se dostáváme k souboru *readIFF.c*, který obsahuje všechny důležité funkce z hlediska načítání dat obrázku. Popíši nyní jednotlivé funkce v pořadí v jakém se vyskytují v tomto souboru:

Funkce *readHEADER()* má za parametr jen identifikační číslo souboru (*fh*), který obsahuje obrázek. Tato funkce nedělá nic jiného, než že přečte prvních 12 bytů

souboru a zkontroluje, zda je to skutečně IFF soubor pro obrázky ILBM. Je-li tomu tak, vrací funkce OK. Jinak vrací ERROR.

Funkce *getChunkID()* má tři parametry. První je zase identifikační číslo souboru (*fh*). Druhý a třetí jsou ukazatele na ID bloku a na jeho délku. Tyto hodnoty dodá funkce, je-li úspěšná. Programátor navyklý na Pascal by napsal, že tyto dva parametry jsou volány referencí. Tato funkce tedy přečte ID bloku a naplní poslední dva parametry jeho typem a délkou.

Funkce *readBMHD()* má také tři parametry, z nichž je zase poslední volán referencí. První je jako obvykle identifikační číslo souboru. Druhý parametr je délka bloku a poslední, parametr je ukazatel na strukturu obsahující hlavní části bloku BMHD. Funkce si přečte blok a naplní strukturu příslušnými údaji.

Funkce *readCMAP()* má opět tři parametry. Jak jste si již určitě všimli, je první znovu identifikační číslo souboru (*fh*). Druhý zase délka bloku a třetí parametr je ukazatel na *UWORD*. Jeho význam nejspíše pochopíte při popisu funkce *main()*, ale teď Vám alespoň řeknu, že představuje ukazatel na mapu barev. Tato funkce přečte CMAP blok a naplní mapu barev příslušnými barvami. Toto se děje na řádkách 129 až 133. Zde bych měl podat krátké vysvětlení pro ty z Vás, kteří nejste příliš zasvěceni do jazyka C. Jak asi už všichni víme, má Amiga 12 bitů na určení jedné barvy - 4 bity na červenou složku, 4 bity na zelenou a 4 bity na modrou. Pro těchto 12 bitů je v mapě barev ale rezervováno celé slovo - 16 bitů. Z toho je zřejmé, že nejvýznamější čtyři bity ve vyhrazeném slovu nejsou použité. Z tohoto důvodu je nutné přeskupit tři načtené byty (červená, zelená a modrá složka) - z toho jsou vždy nejvýznamější čtyři bity nepoužité - do jednoho slova (přesněji - do 12 bitů). A toto se právě děje na výše uvedených řádkách.

Funkce *readBODY()* má pět parametrů. První, jako obvykle, je identifikační číslo souboru. Druhý stále udává délku bloku. Třetí parametr je ukazatel na již otevřenou obrazovku (*screen*). Čtvrtý je ukazatel na strukturu obsahující nezbytné údaje o obrázku a poslední parametr udává komprimaci souboru. Jak si jistě pamatujete z článku v Amiga 1/91, je obrázek uložen po řádkách, a každý bitplane je uložen zvlášť. Zde, podle toho zda je obrázek komprimován nebo ne, je čten příslušný bitplane po řádkách a rovnou je zapsán na obrazovku.

Funkce *readDUMMY()* s dvěma parametry (identifikační číslo souboru a délka bloku) je zde jen proto, že, setkáme-li se s nepodporovaným blokem, tak ho přeskočíme. Tato funkce nemá žádný jiný význam.

A nakonec nám zbyl soubor *imaging.c*, který obsahuje funkci *main()* a pomocnou funkci *Clean_exit()*, která jen zamete všechny stopy. První část souboru, která obsahuje inicializaci různých proměn, by měla být všem známá z předešlých článků. Účelem hlavní funkce je zkontrolovat, že byly načteny všechny nezbytné bloky obrázku a potom obrázek zobrazit.

Přeji Vám hodně úspěchů ve Vašich pokusech s grafikou a doufám, že shledáte tento jednoduchý a tím i nedokonalý prográmek určitým pomocníkem ve Vašem bádání.

POZOR! Číslo řádků jsou uvedena pouze pro lepší orientaci. Proto je ve vlastním zájmu neopisujte!


```

1  /* inclall.h */
2
3  #ifndef INCLALL_H
4  #define INCLALL_H
5
6  #define INTUITION_REV 33
7
8  #include <libraries/dos.h>
9  #include <exec/memory.h>
10 #include <exec/types.h>
11 #include <intuition/intuition.h>
12
13 #include <proto/all.h>
14
15 #endif INCLALL_H
16
17 /* ifftypes.h */
18
19 #ifndef IFFTYPES_H
20 #define IFFTYPES_H
21
22 #ifndef INCLALL_H
23 #include "inclall.h"
24 #endif
25
26 #define ERROR 0
27 #define OK 1
28
29 #define MakeID(a,b,c,d) ((a) << 24 | (b) << 16
30 | (c) << 8 | (d))
31
32 #define ID_FORM MakeID('F','O','R','M')
33 #define ID_ILBM MakeID('I','L','B','M')
34 #define ID_BMHD MakeID('B','M','H','D')
35 #define ID_CMAP MakeID('C','M','A','P')
36 #define ID_BODY MakeID('B','O','D','Y')
37
38 #define COMPR_NONE 0
39 #define COMPR_BYTERUN1 1
40
41 struct BitMapHeader {
42     UWORD w, h;
43     UBYTE nPlanes;
44     UBYTE compression;
45     WORD pageWidth, pageHeight;
46 };
47
48 /* function prototypes */
49 int readHEADER(BPTR);
50
51 int getChunkID(BPTR,
52     LONG *,
53     LONG *
54 );
55
56 int readBMHD(BPTR,
57     LONG,
58     struct BitMapHeader *
59 );

```

```

60 int readCMAP(BPTR,
61     LONG,
62     UWORD *
63 );
64
65 int readBODY(BPTR,
66     LONG,
67     struct Screen *,
68     struct BitMapHeader *,
69     UBYTE
70 );
71
72 int readDUMMY(BPTR,
73     LONG
74 );
75
76 #endif IFFTYPES_H
77
78 /* readIFF.c */
79
80 #include "inclall.h"
81 #include "ifftypes.h"
82
83 readHEADER(fh)
84 BPTR fh;
85 {
86     int count = 12;
87     int actual_count;
88     char *inbuffer;
89     LONG *buffer;
90
91     inbuffer = (char *)AllocMem(count,
92 MEMF_PUBLIC | MEMF_CLEAR);
93     if (inbuffer == NULL) {
94         return(ERROR);
95     }
96
97     actual_count = Read(fh,inbuffer,count);
98     if (actual_count != count) {
99         FreeMem(inbuffer,count);
100        return(ERROR);
101    }
102    else
103        buffer = (LONG *)inbuffer;
104
105    if (((*buffer) != ID_FORM) && (*(buffer+2) !=
106 ID_ILBM)) {
107        return(ERROR);
108    }
109
110    FreeMem(inbuffer,count);
111
112    return(OK);
113 }
114
115 getChunkID(fh, chunkID, chunkLEN)
116 BPTR fh;
117 LONG *chunkID;
118 LONG *chunkLEN;

```

```

41 {
42     int count = 8;
43     int actual_count;
44     char *inbuffer;
45     LONG *buffer;
46
47     inbuffer = (char *)AllocMem(count,
MEMF_PUBLIC | MEMF_CLEAR);
48     if (inbuffer == NULL) {
49         return(ERROR);
50     }
51
52     actual_count = Read(fh,inbuffer,count);
53     if (actual_count != count) {
54         FreeMem(inbuffer,count);
55         return(ERROR);
56     }
57     else
58         buffer = (LONG *)inbuffer;
59
60     *chunkID = *buffer;
61     *chunkLEN = *(buffer + 1);
62
63     FreeMem(inbuffer,count);
64
65     return(OK);
66 }
67
68 readBMHD(fh, chunkLEN, bmhd)
69 BPTR fh;
70 LONG chunkLEN;
71 struct BitMapHeader *bmhd;
72 {
73     int count;
74     int actual_count;
75     char *inbuffer;
76
77     count = chunkLEN;
78     inbuffer = (char *)AllocMem(count,
MEMF_PUBLIC | MEMF_CLEAR);
79     if (inbuffer == NULL) {
80         return(ERROR);
81     }
82
83     actual_count = Read(fh, inbuffer, count);
84     if (actual_count != count) {
85         FreeMem(inbuffer,count);
86         return(ERROR);
87     }
88
89     bmhd->w = *((UWORD *)inbuffer);
90     bmhd->h = (((UWORD *)inbuffer) + 1);
91     bmhd->nPlanes = (((UBYTE *)inbuffer) + 8);
92     bmhd->compression = (((UBYTE *)inbuffer
) + 10);
93     bmhd->pageWidth = (((WORD *)inbuffer) + 8);
94     bmhd->pageHeight = (((WORD *)inbuffer) + 9);
95
96     FreeMem(inbuffer,count);
97
98     return(OK);
99 }
100
101 readCMAP(fh, chunkLEN, colorMap)
102 BPTR fh;
103 LONG chunkLEN;
104 UWORD *colorMap;
105 {
106     int actual_count;
107     int count;
108     char *inbuffer;
109     int nroColorRegs;
110     BYTE *buffer;
111
112     count = chunkLEN;
113     inbuffer = (BYTE *)AllocMem(count,
MEMF_PUBLIC | MEMF_CLEAR);
114     if (inbuffer == NULL) {
115         return(ERROR);
116     }
117
118     actual_count = Read(fh,inbuffer,count);
119     if (actual_count != count) {
120         FreeMem(inbuffer,count);
121         return(ERROR);
122     }
123     else
124         buffer = (BYTE *)inbuffer;
125
126     nroColorRegs = count / 3;
127
128     for ( ; nroColorRegs > 0 ; --nroColorRegs) {
129         *colorMap++ = ( (*buffer++ >> 4) << 8) |
130             ( (*buffer++ >> 4) << 4) |
131             ( (*buffer++ >> 4));
132     }
133
134     FreeMem(inbuffer,count);
135
136     return(OK);
137 }
138
139 readBODY(fh, chunkLEN, scrn, bmhd, compr)
140 BPTR fh;
141 LONG chunkLEN;
142 struct Screen *scrn;
143 struct BitMapHeader *bmhd;
144 UBYTE compr;
145 {
146     int i, j, k;
147     int act_c;
148     BYTE *planes[5];
149     BYTE *scrRow;
150     int bytecount;
151     char *buffer;
152     UBYTE code;
153     BYTE fillbyte;
154     int count = 3;

```

```

157
158  buffer = (char *)AllocMem(count,
MEMF_PUBLIC | MEMF_CLEAR);
159  if (buffer == NULL)
160      return(ERROR);
161
162  for (i = 0; i <= scrn->RastPort.BitMap->Depth
- 1; i++) {
163      planes[i] = (BYTE *)scrn->RastPort.BitMap
->Planes[i];
164  }
165
166  switch( compr ) {
167      case COMPR_NONE:
168          for (i = 0; i <= bmhd->h - 1; i++) {
169              for (j = 0; j <= bmhd->nPlanes - 1; j++) {
170                  scrRow = planes[j] + i*(bmhd->
pageWidth / 8);
171                  act_c = Read(fh, scrRow, bmhd->w / 8);
172              }
173          }
174          break;
175
176      case COMPR_BYTERUN1:
177          for (i = 0; i <= bmhd->h - 1; i++) {
178              for (j = 0; j <= bmhd->nPlanes - 1; j++) {
179                  scrRow = planes[j] + i*(bmhd->
pageWidth / 8);
180                  bytecount = 0;
181
182                  while ( bytecount < (bmhd->w / 8) ) {
183                      act_c = Read(fh, buffer, 1);
184                      code = *((UBYTE *)buffer);
185
186                      if (code 128) {
187                          act_c = Read(fh,
scrRow + bytecount,
188                              code + 1);
189                          bytecount += code + 1;
190                      }
191                      else if (code 128) {
192                          act_c = Read(fh,buffer,1);
193                          fillbyte = *((BYTE *)
buffer);
194
195                          for (k = bytecount;
196                              k <= bytecount + 257/code;
197                              k++) {
198                              *(scrRow + k) = fillbyte;
199                          }
200
201                          bytecount_ += 257 - code;
202                      }
203                  }
204              }
205          }
206      }
207  }
208  break;
209 }
210
211  FreeMem(buffer, count);
212
213  return(OK);
214 }
215
216  readDUMMY(fh, chunkLEN)
217  BPTR fh;
218  LONG chunkLEN;
219  {
220      int actual_count;
221      char *inbuffer;
222
223      inbuffer = (char *)AllocMem(chunkLEN,
MEMF_PUBLIC | MEMF_CLEAR);
224      if (inbuffer == NULL) {
225          return(ERROR);
226      }
227
228      actual_count = Read(fh,inbuffer,chunkLEN);
229      if (actual_count != chunkLEN) {
230          FreeMem(inbuffer,chunkLEN);
231          return(ERROR);
232      }
233
234      FreeMem(inbuffer, chunkLEN);
235
236      return(OK);
237 }
0  /* imaging.c */
1  #include "inclall.h"
2  #include "ifftypes.h"
3
4  void Clean_exit(int);
5
6  struct NewScreen myscreen = {
7      0, /* LeftEdge, must be zero */
8      0, /* TopEdge */
9      320, /* Width */
10     256, /* Height */
11     5, /* Depth, 5 = 32 colors */
12     0, -1, /* DetailPen, BlockPen */
13     0, /* ViewMode, 0 = lores */
14     CUSTOMSCREEN,
15     NULL, /* Font */
16     "MyPaintScreen", /* screen title */
17     NULL, /* no screen gadgets */
18     NULL, /* no CustomBitMap */
19 };
20
21  struct Screen *ms = NULL;
22
23  struct NewWindow mywindow = {
24      0, /* LeftEdge */
25      0, /* TopEdge */
26      320, /* Width */
27      256, /* Height */
28      0, 1, /* DetailPen, BlockPen */
29      0, /* IDCMPFlags */
30      BORDERLESS|ACTIVATE, /* Flags */
31      NULL, /* gadget list */

```



```

32  NULL,/* checkmark */
33  "MyPaint",/* window title */
34  NULL,/* screen */
35  NULL,/* bitmap */
36  320,/* minWidth */
37  256,/* minHeight */
38  320,/* maxWidth */
39  256,/* maxHeight */
40  CUSTOMSCREEN /* own screen */
41  };
42
43  struct Window *mw = NULL;
44
45  struct IntuitionBase *IntuitionBase = NULL;
46  struct GfxBase *GfxBase = NULL;
47
48  BPTR fh;
49
50  struct BitMapHeader bmhd;
51  UWORD *colorMap;
52
53  void main(argc, argv)
54  int argc;
55  char *argv[];
56  {
57      int i;
58      LONG id, len;
59      int nc;
60      int chunkcount;
61
62      chunkcount = 0;
63
64      IntuitionBase = (struct IntuitionBase *)
65          OpenLibrary("intuition.library",
66          INTUITION_REV);
67      if (IntuitionBase == NULL)
68          Clean_exit(0);
69
70      GfxBase = (struct GfxBase *)
71          OpenLibrary("graphics.library",
72          INTUITION_REV);
73      if (GfxBase == NULL)
74          Clean_exit(1);
75      fh = Open(argv[1],MODE_OLDFILE);
76      if (fh == NULL) {
77          printf("name is %s\n",argv[1]);
78          Clean_exit(2);
79      }
80
81      colorMap = (UWORD *)AllocMem(96,
82      MEMF_PUBLIC | MEMF_CLEAR);
83      if (colorMap == NULL) {
84          Clean_exit(2);
85      }
86
87      if ( readHEADER(fh) ) {
88          do {
89              getChunkID(fh, &id, &len);
90              switch( id ) {
91                  case ID_BMHD: {

```

```

89          chunkcount + +;
90          readBMHD(fh, len, &bmhd);
91          myscreen.Width = bmhd.pageWidth;
92          myscreen.Height = bmhd.pageHeight;
93          myscreen.Depth = bmhd.nPlanes;
94          nc = 2 << (bmhd.nPlanes-1);
95
96          ms = (struct Screen *)
97              OpenScreen(&myscreen);
98          if (ms == NULL)
99              Clean_exit(3);
100
101          mywindow.Screen = ms;
102          mw = (struct Window *)
103              OpenWindow(&mywindow);
104          if (mw == NULL)
105              Clean_exit(4);
106          break;
107      }
108      case ID_CMAP: {
109          chunkcount + +;
110          readCMAP(fh, len, colorMap);
111          LoadRGB4(&(ms->ViewPort),
112          colorMap,32);
113          break;
114      }
115      case ID_BODY: {
116          chunkcount + +;
117          readBODY(fh, len, ms, &bmhd,
118          bmhd.compression);
119          break;
120      }
121      default: if ( !readDUMMY(fh, len) )
122          Clean_exit(5);
123          break;
124      }
125      } while(chunkcount != 3);
126  }
127
128  for (i = 1; i < 100000; i + + ) {
129      ;
130  }
131
132  Close(fh);
133
134  Clean_exit(5);
135  }
136
137  void Clean_exit(stage)
138  int stage;
139  {
140      switch(stage) {
141          case 5: CloseWindow( mw );
142          case 4: CloseScreen( ms );
143          case 3: FreeMem( colorMap , 96 );
144          case 2: CloseLibrary( (struct Library *)GfxBase )
145          case 1: CloseLibrary( (struct Library *)Intui-
146          tionBase );
147          case 0: break;
148      };

```

TISKÁRNA NEMUSÍ ZDRŽOVAT

Všimli jste si, že u některých programových disket si počítač před odesláním dat k tiskárně vyžádá znovu bootovací disketu (většinou WB), aby si zde mohl načíst potřebné parametry? Tomuto můžete zabránit zapsáním následujícího řádku do Startup-Sequence:

copy NIL: to PRT:

Pro otevření potřebuje Amiga některé soubory v následujících seznamech:

I/Port-Handler

devs/parallel.device nebo **serial.device**

devs/printer.device

devs/printer/název řadiče

PRT: bude aktivováno i když tiskárna dosud není zapnuta.

(Volně zpracováno podle Amiga M&T)(bb)

FISH-DISKS

Časopis Amiga M&T uveřejnil v čísle 12/90 rozhovor s Fredem Fishem, organizátorem známé PD série "Fish-Disks", která dnes čítá 400 disket. Fred Fish (38) je zaměstnancem firmy Motorola. Zajímá se o vývoj software, hlavně v assembleru. U Amigy si cení rychlou grafiku a multitasking. Měsíčně dostává asi tučt nabídek programů. Pokud to chcete také zkusit, zde je jeho adresa: Fred Fish, East Belmont Drive, Tempe Arizona, 85 284, USA.

AntiDELETE

Určitě se vám to již stalo. V okamžiku, kdy jste právě stiskli klávesu RETURN k vykonání příkazu DELETE, uvědomili jste si, že můžete omylem jiný, pro vás důležitý soubor. V tomto okamžiku nesmíte udělat to, co vás napadlo jako první, ve snaze zachránit co se dá. Nesmíte disk násilně zastavit a disketu vyjmout. Klidně se dívejte, jak disk dokonává dílo zkázy a pak teprve přijde řada na vás. Soubor na disketě se totiž nikdy úplně nesmaže, maže se pouze hlavička, záznam v seznamu souborů. Zde je uložena informace o tom, kde je třeba hledat na disketě ten, který soubor. Smazáním hlavičky v seznamu se označí

prostor, kde je soubor uložen, jako volný pro další záznam. Vlastní data souboru se nemažou, přepíše se pak až dalším ukládaným souborem. Proto nesmíte na disketu po osudném příkazu DELETE již nic ukládat. Jen tak můžete smazaný soubor zachránit.

Zavolejte CLI a pak odešlete příkaz "Diskdoctor?". Potom budete vyzváni k vložení diskety se smazaným programem do disku df0. Až skončí Diskdoctor svoji práci, můžete se podívat do direktoráře (seznamu) příkazem DIR a zde uvidíte i svůj soubor. Nyní zkopírujte všechny soubory na novou disketu a starou disketu znovu zformátujte.

Stejnou úlohu s větší elegancí řeší PD program "Disksofv" na Fish-disku č.164 nebo známý program RECOVER II.

ČÍSLOVÁNÍ ŘÁDKŮ V BASICU

Amiga Basic nezná příkaz RENUMBER, ale s menší oklikou je možné řešit úlohu očíslování řádků basicového programu následovně. Nejdříve je třeba program z Basicu uložit ve tvaru ASCII. To se provede známým způsobem (POZOR! Z důvodů lepší přehlednosti značíme velkými písmeny příkazy, malými písmeny názvy souborů a mezery jsou značeny znakem podtržení "_", který při zápisu samozřejmě musíte vynechat):

SAVE "_navez",A

Opusťte Basic a zavolejte si CLI. Pomocí příkazu TYPE s parametrem "OPT_N" můžete automaticky očíslovat řádky programu "navez". Provede se to tak, že se soubor "navez" zkopíruje do souboru "navez2", ve kterém jsou již řádky očíslovány. Příslušný zápis pro tento postup zní:

TYPE _navez_TO_navez2_OPT_N

Nyní opusťte CLI a opět natáhněte Basic. Z Basicu pak natáhněte soubor "navez2". Všimněte si, že řádky jsou číslovány s krokem jedna. Soubor "navez2" můžete samozřejmě libovolně přejmenovat. Chcete-li výpis programu na tiskárně, pak v CLI zadejte příkaz:

TYPE _navez_OPT_N_TO_PRT:

Věc má jeden háček. Příkaz TYPE nezná formátování do stránek a výpis tedy bude probíhat spojitě (na skládaném papíře půjde tisk přes perforaci). I tento problém lze řešit a to textovým editorem. Nejprve si pořídíte kopii pomocí příkazu:

TYPE _navez_TO_navez2_OPT_N

Očíslovanou verzi programu ("navez2") nyní natáhněte textovým editorem (např. TEXTOMATEM). Program podle libosti zformátujte a pak vytiskněte na tiskárně, jako každý jiný textový soubor.

(Volně zpracováno podle AMIGA M&T) (bb)



SHADOW OF THE BEAST II

■ Ocean 1989

TEST 3/91		
Grafika	97%	Celkem 92%
Zvuk	98%	
Idea	81%	
Přitažlivost	91%	

Postihlo vás veliké neštěstí. Vaše sestra byla unesena zlým Necromancerem, čarodějem mnoha jmen, který je ve vaší rodné zemi Karamoonu obvykle nazýván Zelek. Nejprve musíte zjistit, kam byla vaše sestra unesena a potom se vydáte na zoufalou výpravu, při které máte málo naděje na úspěch. Naštěstí pro vás však v Karamoonu nežijí pouze nepřátelští tvorové (i když ti jsou v převaze, jak jinak). Tak třeba starý Barloom ze západního průchodu či kouzelný stařík z planin ... avšak nepředbíhejme.

Rozdíl mezi prvním BEASTem a BEASTem II je veliký. První díl byl pouze výborně provedenou arkádou, kdežto BEAST II si svým myšlenkovým zpracováním nezadá s kvalitními adventure. Určité části hry (Ishranova továrna či sídlo Goblinů v podzemí) jsou v tomto směru prostě bezkonkurenční. Bude te muset mnoho přemýšlet (i když s naším How-to-Play již pouze minimálně). Provedení nové hry je přitom v podstatě stejně kvalitní, jako tomu bylo u dílu prvního. A jelikož BEAST II je skutečně složitější než jeho předchůdce, je složitější i jeho ovládání, o kterém si musíme něco říci: Za

prvé bych vám radil, abyste tuto hru nehráli na joysticku, nýbrž na klávesnici (kurzorové klávesy pro pohyb a SHIFT pro použití zvoleného předmětu). Hrdina může u sebe mít maximálně 4 předměty. Zvoleným předmětem je takový, u kterého se nachází červená šipka - tou pohybujete pomocí kláves F1-F4. Stisknete-li klávesu "A", můžete komunikovat s druhými postavami a takto získáte mnoho životně důležitých informací. Zeptejte se třeba na SISTER, NECROMANCER atd. Odpověď se bude různit podle toho, s kým a kde mluvíte. Některé osoby nebudou moci na vaši otázku odpovědět, někteří vám podají zkreslené informace a můžete se dozvědět i nějakou tu lež - jako v praktickém životě. Prosívejte tedy zprávy, které obdržíte, a vybírejte z nich to, co potřebujete.

Stisknete-li klávesu "O", nabídnete přítomné osobě předmět, u kterého se nachází šipka. I to je třeba učinit na správném místě. Klávesa "S" vyvolá informace o počtu bodů a sebraných penězích. A to už je asi všechno. Vydejte se tedy vstříc velkému dobrodružství...

Běžte nejprve doleva. Po překonání mostu narazíte na člověka, kterého se pokouší unést nějaká okřídlená potvora. Pokud únosce zlikvidujete, získáte si mužovu vděčnost a pár informací. Jeho skupina, sloužící hodnému drakovi Barloomovi, byla zajata vojáky zlého draka Ishrana, sídlícího ve východní průrvě. Pokuste se osvobodit ubohé zajatce! Nejprve si však všimněte, že muž hovoří o pastech, které vás na cestě k Ishranovi čekají. Zeptejte se ho tedy na pasti (TRAPS) a dozvíte se, že vojáci hovořili buďto o horním (UPPER) nebo dolním

(LOWER) spínači. Toto je voleno náhodně a vy si tuto zprávu musíte dobře zapamatovat! Slezte dolů po kořenech a vydejte se do východní průrvy. Nejprve na vás vyletí jeden okřídlenec a začne pižlat provaz. Musíte ho rychle zneškodnit, jinak se už nedostanete nahoru! Dále musíte vyřídít obra, tlačícího kámen (je třeba vyvinout takový způsob, abyste přišli o minimum energie). Po průchodu dvojité brány z padajících kvádrů se dostanete na svah, po kterém sjíždějí opracované kvádry. Dejte si na ně pozor a zabijte okřídleného dělníka, který se tam nachází. Vypadne z něj klíč. Jděte nahoru a dalšího dělníka nechejte uletět. Jste v Ishranově továrně (kamenolomu). Jděte doleva. Tam pomocí tří pák uchopíte velký kámen, přeneste jej nad bodec a z výšky jej na něj pusťte. Kámen se rozbije a zbyde menší, kterým můžete pohybovat. Avšak co s ním? Vždyť ho pouze můžete shodit do kyseliny. Nechejte tedy zatím kámen kamenem a vyšplhejte nahoru po provazu. Dveře otevřete klíčem, který vypadl z prvního zabitého dělníka. Za okamžik jste u dvojice spínačů. Vzpomeňte si, o jakém spínači hovořili strážci (horní či dolní?). Pohněte tedy touto pákou, nastupte do klece a už jste v hradu.

V hradu jděte až úplně doprava a vezměte ze stolu klíč. Nezapomeňte ještě vyšplhat po jednom z lustrů a z truhly vzít prvních šest penízků (celkově jich potřebujete 36). Vraťte se doleva k prvnímu lustru. Sešplhejte na úplný konec řetězu a skočte ke kleci s tygrem tak, abyste neprobudili stráž. Otevřete klec nalezeným klíčem a potom pohněte dolní pákou. Vraťte se do továrny.

U ostrůvku, na kterém leží kámen, se objeví lávka. Přemístěte kámen na houpačku, vyšplhejte po provazu a seskočte tak, abyste vymrštili kámen nahoru (chce to trochu cviku). Spěchejte rychle do klece, která vás po chvíli vyveze nahoru - kámen zatíží horní klec kladky. Nyní zabijte Ishrana a osvobodte rukojmího, který vám prozradí heslo k Barloomovu sídlu. Vraťte se dolů, jděte doprava a slezte po provazu.

Dříve, než půjdete navštívit Barlooma (vpravo), vydejte se doleva k hospodě (najdete další truhlu a máte 12 penízků!). V hospodě vezměte pivo a opět spěchejte zpátky. U dveří do Barloomova sídla zašeptejte heslo, které vám sdělil vděčný muž u Ishrana, a vstoupíte. Nejprve jděte doprava a vezměte další penízky. Poté vyšplhejte po řetězu nahoru, promluvíte si s Barloomem a vezměte pergamen, který spolu s prstenem stačí k tomu, aby vám "starý muž" vyrobil zbraň na toho neřáda Zeleka. Nyní opět vyšplhejte nahoru a jděte k místu, kde hra začala. Zde pokračujte dále doprava. U visícího provazu vyskákejte doleva nahoru a zabijte potvora, ze které vypadne sekerka. Na stejné úrovni najdete truhlu s dalšími šesti penízky, druhá truhla se nachází o dvě patra výš (vyšplhejte po provazu). Nyní jděte doleva a seskočte dolů. Měli byste mít 30 penízků. Vraťte se na místo, kde je země pouze tenkou skořápkou. Vyskočte nahoru tak, abyste se při dopadu na zem propadli do jeskyně.

V jeskyni najdete poslední truhlu, máte nyní 36 penízků. Utíkejte doleva, těsně sledujte prchající stráž a sekerkou zlikvidujte chlapíka, který sedí u rumpálu a stahuje most. Poté zabijte všechny ostatní a standardní zbraní zničte dveře. Vyšplhejte se po lustru nahoru, přepněte spínač u druhého lustru (dočasně vypíná bezpečnostní zařízení) a běžte rychle doleva, kde musíte přehozením páky spustit další lustr. Rychle se vraťte zpět! Nyní jděte doprava, slezte dolů, pobijte strážce, vezměte vpravo na stole ležící klíč a nechejte se chytit do klece. Dostanete se do vězení. Strážci nabídnou pivo a vyvráťte dveře. Rychle vyšplhejte po lustru a zničte nahoře hlídajícího muže dřív, než stačí zlikvidovat vašeho spoluvězně. Vraťte se dolů, zabijte opilého strážce a vezměte jeho klíč. Tím otevřete dveře celý druhého vězně, který se nachází o patro výš. Vyšplhejte nyní úplně nahoru, jděte doleva, pobijte strážce a vezměte prsten. Nyní jděte doprava ke dveřím, jimiž marně lomcují oba zachránění, a otevřete je

posledním zbylým klíčem. Nechejte oba vězně běžet před vámi, neboť oni se stanou obětí bezpečnostního zařízení a vy vyváznete se zdravou kůží (tomu se říká tvrdá realita!). Vyjděte z jeskyně opět na povrch.

Nyní se vydejte doprava. Překonejte řeku a překročte most. Těsně za mostem narazíte na dalšího neřáda, ozbrojeného mečem. Nezbíjejte ho, ale couvejte a vylákejte ho na most, kde se propadne. Pokračujte ve své cestě vpravo a narazíte na starouška. Dejte mu prsten a pergamen a on vám vyrobí zbraň, která může zničit Zeleka. Nyní se vraťte zpět doleva, propadněte se mostem a...

Země, ve které jste se ocitli, patří do jiného světa. Najděte roh (roh foukací) a koukejte se rychle dostat do Karamoonu - domů vás zaveze veliký šnek za 36 zlatých penízků (vyděrač jeden, ještě že jste spořili!). Tak, jste opět v Karamoonu. Jděte doprava, překonejte bažinu (zde se hodí nová zbraň) a na břehu moře zadujte na roh. Připluje loďka-obludka, která vás převezme až k sídlu zlého Zeleka. Co máte dělat tady, to už je vámdoufám zcela jasné. Pomocí zbraně odkouzelného děduláše zlikvidujte Zeleka a vaše sestra bude zachráněna!

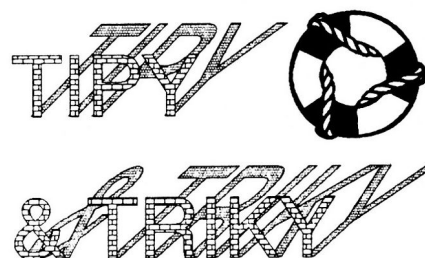
SHADOW OF THE BEAST II je rozhodně to nejlepší, co firma Psygnosis zatím vyrobila. A jelikož **BEAST II** má poněkud otevřený konec, můžeme se určitě těšit na **BEASTa III**!

NÁZOR MARTIN:

Musím říct, že hra **SHADOW OF THE BEAST II** mě opravdu nadchla. Výborná grafika a zvuk spolu s dobrým nápadem a správnou motivací svědčí o tom, že se jedná o dokonalý produkt fy Psygnosis, která si už tak získala mnoho svých příznivců.

NÁZOR TOMÁŠ:

SHADOW OF THE BEAST II považuji za jednu z nejlepších her roku 1990. Firma Psygnosis mě opět jednou překvapila. Hra má vynikající provedení, o nic horší námět a výbornou atmosféru. Jedná se asi o nejvydařenější syntézu arcade a adventure vůbec.



BATMAN THE MOVIE: Na úvodní obrazovce napište JAMMMMM. Obrázek se otočí vzhůru nohama a Batman získá nekonečný počet životů. Klávesou F10 se můžete dostat do další soustavy.

E-MOTION: Na titulním obrázku napište MOONUNIT a stiskněte Return. Poté hru spusťte. Klávesou F1 se dostanete o level dále, klávesou F2 o level zpět. Klávesa F3 vás přenese o deset levelů dále, klávesa F4 o deset levelů zpět.

BEACH VOLLEY: Nejprve zapněte pauzu, potom napište DADDY BRACEY a opět pauzu opusťte. Nyní se klávesou F1 můžete dostat do dalších levelů.

STORMLORD: Když se na obrazovce objeví jméno programátora a grafika, napište DRAGONBRIDGE. Po spuštění hry se stisknutím klávesy "L" dostanete do dalšího levelu.

BACK TO THE FUTURE 2: Klávesou "P" zapněte pause-mód a pak napište THE ONLY NEAT THING TO DO (pozor, německá klávesnice: Y = Z!). Získáte nekonečný počet životů a klávesou "Y" se přenesete do dalšího levelu.

ROLLER COASTER RUMBLER:

Level 2:	AAAGGG
Level 3:	ALIENS
Level 4:	COFFEE
Level 5:	ZARNIE
Level 6:	FRIGHT
Level 7:	TERRER

KARATE KID II: Pomocí klávesy "P" se můžete dostat do dalšího levelu až ke konečnému souboji.



3. díl

amiga 3/91

blízko, ponořte se, abyste viděli cíl. Sledujte přímku, protínající cíl. Odsopoda po ní začne stoupat kolečko, nazývané CCIP-ukazatel. Jakmile toto kolečko překryje zameřovací čtvereček (cíle), vypusťte bombu (spoušť nebo Space). Vyrovnajte svůj letoun, abyste se nezapíchlí do země, a naberte výšku, abyste vyklidili místo dopadu bomby.

HUD raket AGM-65B Maverick

S touto zbraní je skutečně radost pracovat. Mavericky mají ve špičce-zamontovanu TV-kameru s funkcí ZOOM, jejíž obraz je přenášén do COMEDu. Zaměření cíle probíhá stejně jako u bombardovacího HUDu. Je-li cíl dobře zaměřen, počkejte si, až budete dostatečně blízko (IN RNG) a dalšímstisknutím spouště raketu vypusťte. Zafixování zrušíte klávesou "X".

HUD kanónu M61-A1

Váš kanón může být použit jak v HUDu vzduch-vzduch, tak v HUDu vzduch-země. Rozdíl je takový, že při střelbě na pozemní cíle je třeba cíl zafixovat, jako tomu bylo u obou předchozích HUDů. Je třeba letět nízko a ve vhodnou chvíli se sklopit, vystřelit, a opět nabrat výšku.

Přistávací HUD ILS

Tento HUD zapnete klávesou F7. Princip systému ILS je jednoduchý. Z přistávací dráhy je promítnut paprsek, který je snímán vaším letounem. Tento paprsek určuje nejlepší dráhu přistání. V HUDu se nachází horizontální a vertikální přímk, jejichž poloha se mění dle vaší polohy vůči paprsku. Tvoří-li obě přímk přesný kříž, letíte správně. Je třeba též koordinovat rychlost a náklon pomocí systému AOA. A ještě malé upozornění: Pro přistáníse systémem ILS je na vašem letišti vybavena pouze dráha #9.

(Pokračování příště)

NOVINKY + ZAJÍMAVOSTI

* Firma Domark vstoupila na trh s novým atraktivním leteckým simulátorem. Proč je tento simulátor tak atraktivní? Umožňuje totiž hráči létat se sovětským stíhačem No. 1MiG-29 Fulcrum. Předběžné testy dopadly uspokojivě.

* Francouzská firma Delphine Software, která se proslavila svými adventure FUTURE WARS a OPERATION STEALTH, dokončuje novou adventure snázvem CRUISE FOR A CORPSE. První uveřejněné fotografie vypadají velice slibně.

* Rings Of Medusa, hra, která se stala v Německu v loňském roce jednou z nejprodávanejších, se dočkala pokračovatele. Firma Starbyte jej uvádí na trh pod jménem THE RETURN OF MEDUSA a dle reklam se mají fanoušci strategických her na co těšit.

* Oblíbenost her se pozná nejlépe podle toho, kolik jich bylo v tom kterém měsíci prodáno. Uvádím žebříček nejprodávanejších her u firmy Rushware (jeden z největších distributorů herního software v Německu) pro měsíc únor:

1. Powermonger
2. The Secret of Monkey Island
3. Indianapolis 500
4. Buck Rogers
5. Wing Commander
6. M.U.D.S.
7. Ultima VI
8. Champions of Krynn
9. Captive
10. Wonderland

* V herním světě se objevil neočekávaný velikán. Je jím nová strategická hra od firmy Psygnosis s názvem LEMMINGS. Hráčovým úkolem je dostat co největší počet roztomilých Lemmingsíků z jednoho místa vygenerovaného světa (těch bude pěkných pár) na druhé. Svět je však plný nástrah a hráč musí zamezit ztrátám ve své populaci - musí do cílového bodu dovést předem určenou část populace v předem určeném čase.

Zdá se vám to trochu zmatené? Nic si z toho nedělejte. Musíte si počkat na podrobnější popis, co do složitosti a atmosféry si LEMMINGS totiž nijak nezadáji s populárními Populous.

* Firma Maxis, která se proslavila strategickou hrou Sim City, neusnula na vavřínech. V únoru tohoto roku se objevila nová strategická hra (zatím pouze ve verzi pro Maca) SIM EARTH. Všichni testéři jsou nadšení, takže se můžeme těšit na brzký příchod Amiga-verze.

* Firma Gremlin Graphics dosáhla velikého úspěchu s akční hrou LOTUS ESPRIT TURBO CHALLENGE. Hodlá z tohoto úspěchu rychle těžit a dvě motoristické novinky z jejich dílny se jmenují TOYOTA CELICA GT RALLY a TEAM SUZUKI.

* Populární Sentinel si našel svého pokračovatele. Je jím strategická hra HARD NOVA.

* Svět počítačových fantasy her je opět o něco bohatší. Nový kvalitní přírůstek pochází od firmy Sir Tech a jmenuje se BANE OF THE COSMIC FORGE. První testy dopadly výborně.

* Firma Sierra On-Line vypustila na trh (po krátké přestávce) některé z očekávaných her (zatím pouze MS-DOS verze). Jedná se o hry KING'S QUEST V, HERO'S QUEST II, OIL'S WELL a JONES.

* Novou vysoko hodnocenou akční hrou je SPINDIZZY WORLDS. Zdatný herní fanda již podle názvu pozná, o co se asi tak jedná. Výrobce je firma Activision, hotovy jsou zatím verze pro Amigu a Atari ST.

* Firma Spectrum HoloByte, autor ohromně úspěšného simulátoru F-16 FALCON, vstupuje na trh s novinkou, která se jmenuje FLIGHT OF THEINTRUDER. MS-DOSová verze tohoto leteckého simulátoru si získala výborná hodnocení, již reklama spojuje nový program s kvalitami FALCONa; máme se tedy na co těšit.

KING'S QUEST

■ Sierra On-Line 1986

TEST 3/91

Grafika	43%	Celkem 51%
Zvuk	14%	
Idea	69%	
Přitažlivost	75%	

KING'S QUEST je první ze (zatím)pětidílné série King's Questů, adventure, nesoucí značku firmy Sierra On-Line. Děj nás zavádí do Daventry, malého království, jako vystřiženého z pohádkové knížky. Vžijte se do role sira Grahama, dobrodruha, jehož hlavu nikdy neopustila později tak slavná čepice, a proveďte jej množstvím dobrodružství, abyste pro sebe získali korunu.

Sir Graham byl povolán ke králi. Veďte jej tedy nejprve do zámku. Zde se dozvíte, že král Eduard Benevolentní očekává svůj konec a hledá nástupce. Pokud dokážete zeslabého království udělat opět mocnou říši, stane se novým králem sir Graham. K tomu však budete potřebovat tři

magické předměty, které se nacházejí někde na území království. Jedná se o magické zrcadlo, které může předpovídat budoucnost a umí i další zajímavé věci, pak je to kouzelná truhla, vždy plná zlatáků, a konečně kouzelný štít, který ochrání jeho majitele před vším zlem. Vydejte se tedy na cestu.

Ze zámku jděte doleva. V sektoru C4 odvalte kámen (MOVE ROCK) a vezměte dýku (TAKE DAGGER). Jděte ke studni v sektoru A5 a vlezte do ní (ENTER WELL). Skočte do vody, plavte a pak se ponořte (DIVE). Vplujte do díry, kterou vidíte nalevo. Tam se nachází drak, strážce zrcadla. Přiblížte se k němu a hoďte po něm dýku (THROW DAGGER). Vezměte zrcadlo, vraťte se stejnou cestou, jakou jste přišli, a vyšplhejte po laně ze studně (CLIMB ROPE).

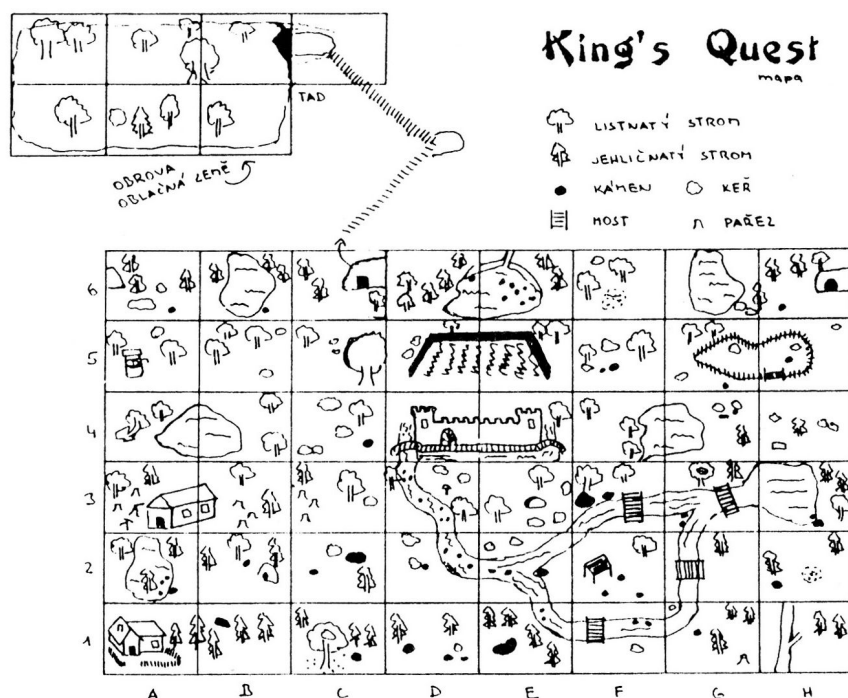
Nyní se vypravíme za druhým předmětem. Buďte jím kouzelný štít. Nejprve jděte do sektoru C5 a vyšplhejte na dub (CLIMB TREE). Vezměte zlaté vejce (TAKE EGG) a slezte dolů. Jděte do sektoru F6 a utrhněte čtyřlístek (TAKE CLOVER). Potom se přesuňte do

sektoru H6. Jakmile přiletí pták, postavte se na střed obrazovky a ve vhodný okamžik vyskočte (JUMP) a chytněte se ptáka. Spadnete na jinak nepřístupné straně divoké řeky Raging River. Seberte muchomůrku (TAKE MUSHROOM) a seskočte do díry v sektoru E3. Kryse v chodbě dejte zlaté vejce (GIVE EGG) a otevřete dveře. U trůnu krále skřítků vezměte štít (TAKE SHIELD) a odejděte doleva. Zde snězte muchomůrku (EAT MUSHROOM) - zmenšíte se a snadno proklouznete ven.

Ano, teď již musíte objevit pouze kouzelnou truhlu. Jděte do sektoru C1 a vezměte ořech (TAKE WALNUT). Otevřete jej a zjistíte, že je zlatý. Nyní jděte k některému z mostů, hlídaných trollem (sektory F1, G2 a F3). Dejte mu ořech (GIVE WALNUT) a on vás pustí na ostrov. Zde si popovídejte se starým gnomem (TALKGNOM). Dostanete hádanku - uhodnout jeho jméno. Nepodaří se vám to a obdržíte klíč (TAKE KEY). Jděte ke vchodu do hory v sektoru C6 a odemkněte dveře klíčem (UNLOCKDOOR). Jděte nahoru. Jste-li již úplně nahoře, jděte doleva. Zde se nachází obr, který drží truhlu. Rychle se schovejte za strom a počkejte, až obr usne. Pak mu opatrně seberte truhlu (TAKE CHEST) a vraťte se opět dolů.

Nyní jděte ke králi. Pokloňte se mu (BOW KING). Král vás uvítá, předá vám vládu a zemře. Nechť žije král Graham!

Pokud jste hru hráli tak, jak byla zde popsána, zarazíte se teď ařeknete: "No jo, ale co tybody? Vždyť mám sotva polovičku z maximálně dosažitelného skóre." Ano, to je pravda. Existuje totiž mnoho nepovinných akcí, za které můžete body získat. Kromě toho každý ze tří předmětů lze získat i odlišným postupem. Např. drak, hlídající zrcadlo, může vytlačit velký kámen ve strachu před vodou. Pokud jste ugnoma a jako jeho jméno uvedete "IFN KOVHGROGHPRM", neobdržíte klíč, nýbrž kouzelné fazole (BEANS). Zasadíte-li je do země v sektoru H2, tak po vyrostlé fazoli se dostanete do obrovské země. V sektoru D1 najdete misku (BOWL). Naplňte



ji vodou (FILLBOWL) a doneste ji do dřevorubcovy chaty v sektoru A3. Na oplátku si můžete vzít housle (FIDDLE). V sektoru D6 můžete též narazit na elfa, který vás obdaruje kouzelným prstenem. Setkáte-li se v sektoru F5 s dobrou vílou, jděte do chalupy čarodějnice (sektor A1), kde se povaluje sýr (švýcarský!), schovaný v kredenci (OPEN CABINET). V sektoru D2 lze najít kamínky (PUBBLES), které se vám mohou hodit u obra.

Avšak ani po splnění všech těchto nepovinných úkolů nemáte plný stav bodů. Přijdete-li tedy ještě na něco dalšího, prosím, napište mi.

Starší vývojový systém fy Sierra On-Line obsahuje několik standardních funkcí, které jsou většinou obsaženy ve všech hrách tímto systémem zpracovaných. Zde jsou nejobvyklejší kombinace a funkce:

F1 nebo HELP: Informační obrazovka

F2: Zapíná/vypíná hudbu (pípání)

F3: Zopakuje poslední zadaný příkaz

F5: SAVE pozice

F7: LOAD (RESTORE) pozice

F9: Nová hra

Zbylé funkční klávesy mívají speciální funkce (např. u LARRYho v kasínu).

ESC: Pauza

TAB: Inventář

CTRL + J: Přepíná ovládání na joystick.

K ovládání smělého dobrodruha (později krále) Grahama je nejlepší použít klávesnici a to nikoliv kurzorové klávesy, nýbrž číselný blok (to platí pro všechny hry fy Sierra). Tam můžete totiž zadávat i chůzi šikmo (klávesy 1, 3, 7 a 9), klávesou 5 se sir Graham zastaví. K překonání jezer a rybníků použijte SWIM, k chycení se ptáka JUMP, k úniku před čarodějnici DUCK. Dejte si pozor i na další darebáky, kteří vás chtějí připravit o život (kouzelník, vlk, skřet). Hodně štěstí a dobrou zábavu.

DUNGEON MASTER

3. podlaží

Abyste mohli projít třetí podlaží, musíte najít šest klíčů.

1) Komnaty strážce. Poprvé se zde setkáte s modrými skřety. Z výklenků musíte vytrvalou teleportací vyzískat truhlu. Vezměte zrcadlo a ukažte jím na oko ve stěně - naleznete (mimo jiné) první klíč.

2) Klenba. K otevření prvé dvojice dveří musíte obětovat penízky. Pootevření obou tajných průchodů získáte klíč.

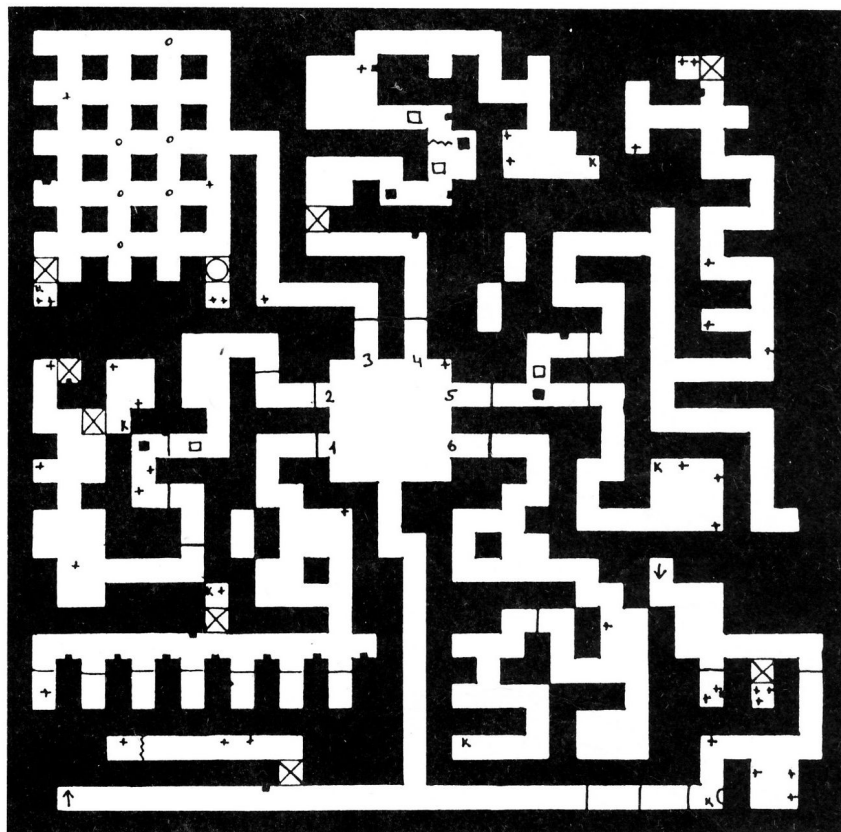
3) Matrice. Zde se poprvé setkáte s falešnou stěnou (vezměte předměty). Pozor na magické body, pokud na ně vstoupíte, podlehnете

kouzlu matrice! Z místnosti za tajnými dveřmi vezměte třetí klíč.

4) Čas je podstatou. Po stisknutí prvního spínače se na chvíli otevře tajný průchod. Musíte být rychlí, abyste jím prošli. U otevřené jámy stiskněte spínač a hodte nějaký předmět na clonu - jáma se uzavře. Další jáma - stiskněte spínač a rychle běžte dozadu (pozadu, he he!). Poslední spínač odruší všechny překážky. Vezměte čtvrtý klíč.

5) Komnata gemu. Nalezený gem vložte do štěrbinu u zavřených dveří a vezměte pátý klíč.

6) Jeskyně monster. Po zničení potvor vezměte klíč. Za dveřmi najdete první RA-klíč. Nové kouzlo: DES + EW zhmotní nemateriální věci (na duchy apod.). Se všemi klíči je cesta do čtvrtého podlaží hračkou. Chaosovi zmar!



TAD

↑ schody nahoru

} silová stěna

⊗ tajný průchod

⊙ falešná stěna

■ podlahový spínač

(oživovací výklenek

↓ schody dolů

† spínač

κ klíč

+ předmět

□ propadlo

o magický bod

transgas *NABÍZÍ:*

TRANSGAS je tu pro Vás

Pro řešení problematiky proudění plynu a jiných médií v potrubních systémech nabízíme:

- znalosti získané 18-letou zkušeností v oblasti vývoje metod simulace, optimalizace, sběru dat, výpočtů pro vyhodnocení provozu i jejich úspěšné aplikace na soustavě tranzitních plynovodů v ČSFR i v zahraničí,
- výkonný „software package“ pro simulaci proudění plynu SIMONE v ČSFR i v zahraničí (SRN, Maďarsko, Dánsko atd.),
- vlastní software typu SCADA pro sběr dat a řízení rozsáhlého technologického systému,
- programy pro optimalizaci provozních nákladů a dalších parametrů, optimalizaci provozních postupů atd.,
- veškerý aplikační software pro využití při dopravě plynu a jiných médií,
- aplikace zabezpečíme podle vašich požadavků dodávkou obecného software nebo dodávkami „na klíč“, náš software zařídíme do vašeho dosavadního systému, vyvineme vhodnou metodu pro řešení vašeho problému, poskytneme konzultaci.

PIŠTE, VOLEJTE, NAVŠTIVTE NÁS!



Naše adresa zní: Tranzitní plynovod, k. p., Praha
Odbor technických služeb a zakázek
Štěpánská 28
113 94 Praha 1
tel. č.: 83 97 87
82 93 40