

Mesačník pre užívateľov počítačov AMIGA

AMIGA

star

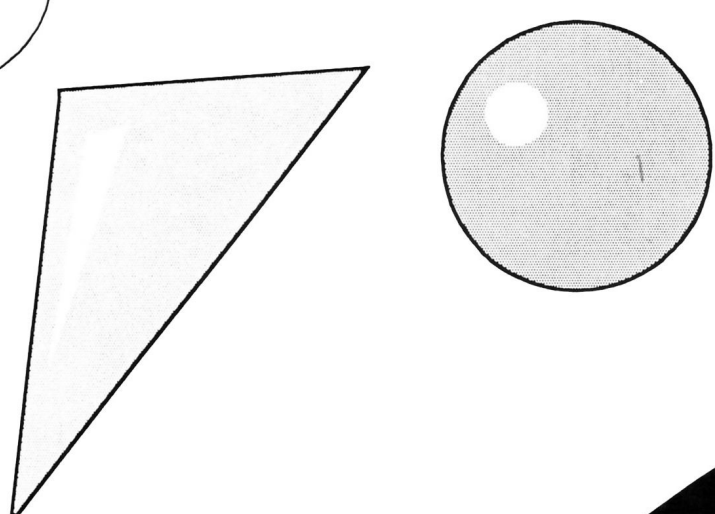
ročník 1.

číslo 5.

máj 1992

cena 19,— Kčs





GURU

Floppy Magazine

Vážení priatelia !

Redakcia časopisu **AMIGA star** a **GURU** vás srdečne pozýva na stretnutie užívateľov počítačov Commodore **AMIGA**. Akcia nesie názov :

Re-Setkání jaro '92

a uskutoční sa vo veľkej sále podniku Vlněna Brno na Přízovej ulici č. 3

dňa 30. mája 1992 v čase od 9:00 do 18:00 hod.

Jedná sa o pokračovanie už známych a úspešných stretnutí. Akcia nie je určená výlučne pre majiteľov **AMIG**, s radosťou uvítame aj širokú verejnosť - záujemcov o počítače **AMIGA**. Budú predvedené možnosti počítačov **AMIGA 500, AMIGA 3000, CDTV**, počítačová grafika, hry...

Na stretnutí budú prítomní zástupcovia československých firiem, zaoberajúcich sa výrobou alebo distribúciou software a hardware. Akcia bude spojená s predajom softu, hardware, časopisov a literatúry.

V sále budú k dispozícii stoly s elektrickými prípojkami v obmedzenom počte, preto vám ponúkame možnosť rezervovania stolov na našej adrese.

Nezabudnite si vziať so sebou počítač, monitor a el. rozdvojky.

Vaše prípadné otázky posielajte na adresu:

AMIGA star
Kvapilova 792
66601 Tišnov,

alebo :
AMIGA star
Hurbanovská 64
94656
Dulovce,

Cena vstupenky je 15,- Kčs, stôl 25,- Kčs

prípadne volajte tel. číslo: 0504/793.

Tešíme sa na vašu účasť,
s pozdravom

Redakcia A-star a GURU

Redakcia časopisu AMIGA star vyhlasuje konkurz
na obsadenie miesta:

Odborný redaktor

Podmienky: vzdelanie min. SŠ, aktívna znalosť
nemčiny alebo angličtiny, prax v oblasti obsluhy DTP
programov a programovania Commodore AMIGA.
Výhodné platové podmienky. Písomné prihlášky
doplnené stručným životopisom posielajte na adresu
redakcie (fax:0818/85 36).

Ďalej hľadáme externých spolupracovníkov.

Počítačová grafika

V dnešnej časti kurzu počítačovej grafiky vás oboznámim so základmi matematického aparátu, ktorý budeme potrebovať pre ďalšiu prácu. Jedná sa iba o zopakovanie stredoškolskej matematiky - základov vektorového počtu a analytickej geometrie. Potom sa dáme do roboty a vytvoríme prvé obrázky.

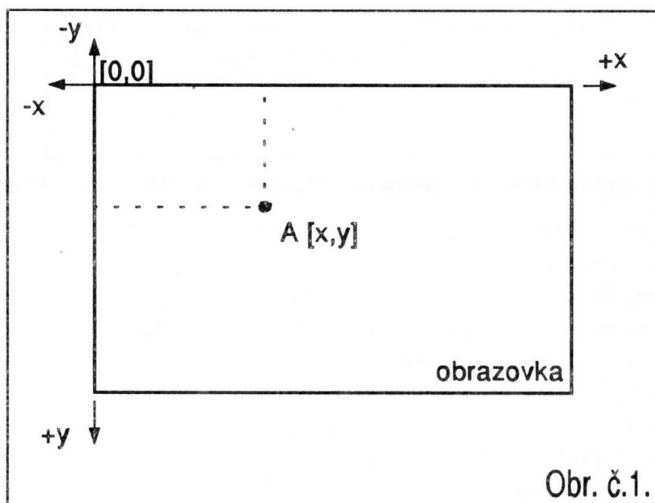
Súradnicový systém

Monitor počítača predstavuje vlastne rovinu, na ktorú budeme kresliť. Aby sme sa vedeli orientovať, zavoláme si na pomoc pravouhlý súradnicový systém. Vodorovný smer vyznačuje os X a zvislý smer os Y. Počiatok súradnicového systému (origo) umiestnime do ľavého horného rohu obrazovky. Teraz už každý bod monitoru môžeme jednoznačne popísať súradnicami (x,y) [obr.1]. Vzhľadom k tomu, že sme origo umiestnili do ľavého horného rohu obrazovky, hodnoty X ako aj Y môžu byť len kladné čísla. V prípade záporných čísel by bod ležal mimo obrazovku.

Na určenie pozícií bodu na obrazovke používame aj vektory. Vektor je definovaný súradnicami (x,y), ale zároveň udáva aj smer: úsečka, ktorá vznikne spojením origa a bodu o súradniciach (x,y) určuje smer vektora. Vektor okrem smeru je charakterizovaný aj veľkosťou, čo je vlastne vzdialenosť bodu A[x,y] od origa. Podľa Pytagorovej vety ľahko vypočítame veľkosť vektora:

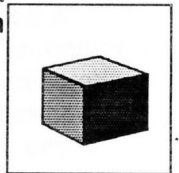
$$|x,y| = \text{SQR}(x^2 + y^2)$$

(pozn. funkcia SQR vracia druhú odmocninu)



Na popísanie trojdimenzionálnych obrázkov budeme potrebovať ešte jednu súradnicu - hĺbku, ktorú označíme "z". Každý bod priestoru je potom určený tromi súradnicami (x,y,z), resp. vektorom. Veľkosť vektora potom vypočítame analogicky:

$$|x,y,z| = \text{SQR}(x^2 + y^2 + z^2)$$



Vektory môžeme sčítať, odčítať, násobiť, ale k tomu sa ešte vrátíme.

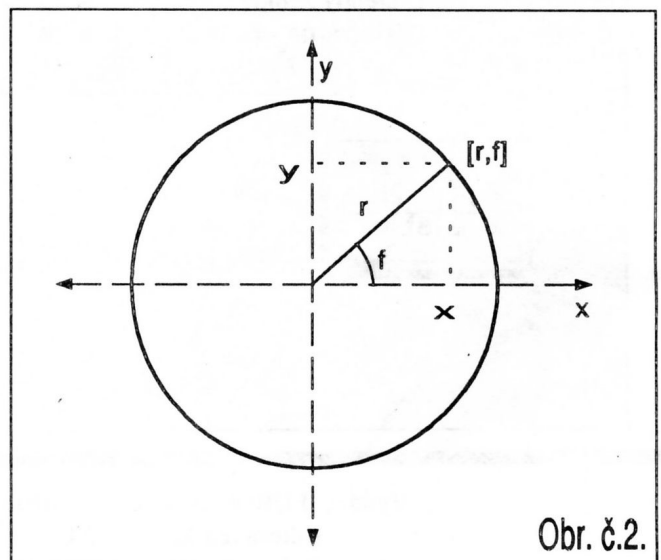
Budeme používať aj tzv. polárny súradnicový systém. Súradnica bodu je daná jeho vzdialenosťou ("r") od počiatku a uhlom ("f"), ktorý zvierajú spojnica bodu s origom a os X. Súradnica bodu je potom daná dvojicou (r,f) - viď. obrázok č.2.

Prepočet do pravouhlého súradnicového systému je jednoduchý, stačí si zopakovať definície funkcií sínus a kosínus:

$$\sin(f) = y/r \quad \cos(f) = x/r$$

Z nich dostaneme:

$$x = r \cdot \cos(f), \quad y = r \cdot \sin(f)$$



Funkcie, krivky

Ak matematicky chceme popísať tvar nejakej krivky, musíme nájsť vhodnú funkciu. Ovšem nesmieme zabudnúť, že každému bodu x prislúcha

iba jedna hodnota y . Prakticky to znamená tolko, že nikdy nemôžeme dostať dva body zvisle nad sebou. Preto napríklad pomocou funkcie nemôžeme popísať kružnicu. V tomto prípade potrebujeme dve funkcie:

$$x=r\cos(f), y=r\sin(f)$$

Hodnota "r" udáva polomer a "f" je parametrom funkcií, ktorý sa mení od 0 po 360 stupňov (0 až 2π).

Podobne ani elipsu, alebo špirálu nemôžeme popísať jednou funkciou, potrebujeme ich viac.

Grafické príkazy

Ako už bolo spomenuté, programovať budeme v AMIGA Basicu. Skúsenejší programátori uvedené algoritmy alebo svoje nápady môžu zrealizovať aj v iných programovacích jazykoch.

Prv než pristúpime k naprogramovaniu nášho prvého algoritmu, musíme si objasniť niektoré grafické príkazy a funkcie AMIGA Basicu. Jedným z najdôležitejších parametrov určujúcich grafický výkon počítača je rozlíšenie. Rozumie sa ním počet bodov, ktorý dokáže zobrazíť v smere X a Y. Pritom veľmi významný je aj počet zobrazených farieb. Naša AMIGA bola pôvodne koncipovaná ako hrací automat, kde veľmi významnú úlohu zohráva práve grafika. Grafický výkon AMIGY je relatívne vysoký vzhľadom k tomu, že je to domáci počítač. Maximálne rozlíšenie je 640 x 512 bodov pri použití 16 farieb (Hi-Res Interlace), ktoré si môžeme vybrať z palety 4096 farieb. Ako už viete, pri rozlíšení 512 vodorovných riadkov je obraz nestály, nepríjemne bliká. Preto sa prakticky využíva iba režim 640 x 256 bodov (Hi-Res NoInterlace), alebo nižšia rozlišovacia schopnosť - 320 x 256 bodov (Low-Res). V tomto móde je dokonca možné zobrazíť až 32 farieb. Najčastejšie budeme používať tento grafický mód. Zámerne nie sú spomenuté ostatné grafické módy AMIGY, lebo nemajú väčší praktický význam (HAM), alebo ich ešte pomocou vyšších programovacích jazykov nevieme využiť (Super Hi-Res, Dynamic HAM a pod.).

Na otvorenie novej obrazovky slúži príkaz SCREEN. Jeho prvým parametrom je číslo, ktoré slúži na označenie obrazovky. Potom nasleduje šírka a výška obrazovky zadaná v bodoch. Ďalší parameter určuje počet bitových polí, čiže počet farieb. Počet farieb sa vypočíta ako číslo 2 umocnené na počet bitových polí. Posledný parameter definuje grafický mód. My budeme pracovať v móde Low-Res, preto príkaz pre otvorenie obrazovky bude vyzeráť nasledovne:

```
SCREEN 1,320,256,5,1
```

Podrobnejšie informácie o jednotlivých príkazoch AMIGA Basicu sa môžete dozvedieť z manuálu, ktorý je súčasťou základnej dodávky počítača.

Ak už máme obrazovku otvorenú, môžeme do nej kresliť. Príkaz, ktorý vykreslí jeden bod na obrazovke má syntax:

```
PSET (x,y),F
```

x,y predstavujú súradnice bodu a F udáva číslo registra farby.

Na kreslenie úsečky slúži príkaz LINE:

```
LINE (x1,y1)-(x2,y2),F
```

kde x1,y1 určujú súradnice počiatočného bodu a x2,y2 súradnice koncového bodu. F tu tiež udáva číslo registra farby. Uvedené dva základné príkazy už nám postačujú k tomu, aby sme mohli nakresliť jednoduché obrázky. Ale ešte predtým si musíme niečo povedať o farbách.

V móde Low-Res máme k dispozícii 32 farbových registrov. Do týchto registrov si môžeme namiešať (nadafinovať) ľubovoľné farby pomocou príkazu PALETTE. Pri kreslení sa potom už len odvolávame na číslo registra farby.

Farby sa vytvárajú miešaním troch základných farieb: červenej (R), zelenej (G) a modrej (B). AMIGA umožňuje nastavovať intenzitu týchto zložiek v 16 stupňoch a ich kombináciami dostaneme spomínaných 4096 farieb. Ak je intenzita všetkých zložiek maximálna, výsledná farba bude biela; ak je ich intenzita nulová, výsledok bude čierna farba.

Príkaz PALETTE má nasledovný syntax:

```
PALETTE FR,R,G,B
```

FR predstavuje číslo farbového registra. R,G,B sú tiež čísla, ktoré definujú percentuálny podiel základných farieb vo výslednej farbe. Môžu mať hodnoty v rozsahu 0.00 až 1.00.

Máme tu ešte jeden problém: AMIGA Basic nedokáže kresliť na obrazovku, ale iba do okna. Preto musíme na obrazovke najprv otvoriť okno, do ktorého potom môžeme kresliť. To docielíme príkazom WINDOW:

```
WINDOW CO,"Titul",(X1,Y1)-(X2,Y2),T,OB
```

CO - číslo okna.

Titul - meno okna, ktoré bude vypísané v hornej lište.

X1,Y1, X2,Y2, súradnice ľavého horného a pravého dolného rohu okna; ak nie sú zadané, vytvorené okno prekryje celú obrazovku.

T - číslo udávajúce spôsob manipulácie s oknom.

OB - číslo obrazovky, na ktorej sa má okno vytvoriť.

Prvý obrázok

Vyzbrojení základnými poznatkami AMIGA Basicu a matematiky sa teraz dajme do roboty. Na začiatok to bude obyčajná kružnica vykreslená bod po bode. Áno, už počujem vaše námietky: "prečo práve kružnicu a to ešte k tomu z bodov, keď AMIGA Basic má na to príkaz CIRCLE?". Odpoveď: na jednoduchom príklade sa učí najľahšie a ako uvidíte, modifikáciou algoritmu kreslenia kružnice môžeme dostať veľmi zaujímavé obrázky. Pri kreslení budeme využívať iba jeden "grafický" príkaz: vykreslenie bodu - PSET.

Ako sme si už v stati o funkciách a krivkách povedali, na vykreslenie kružnice potrebujeme dve funkcie: sínus a kosínus. Obe funkcie vracajú hodnotu v rozsahu -1 až +1 a priemer kruhu by bol iba 1 bod. Preto hodnoty funkcií vynásobíme číslom R, ktorý môžeme považovať za polomer kružnice. Počiatok súradnicového systému v AMIGA Basicu je štandardne ľavý horný roh. Preto po vykreslení kružnice by sme videli len jeho štvrtinu, ostatné časti by boli mimo obrazovku. Problém vyriešime tak, že stred kružnice posunieme do stredu obrazovky. Ak si otvoríme Low-Res obrazovku t.j. o rozlíšení 320x256 bodov, potom stred obrazovky bude mať súradnicu 160,128. K vypočítaným hodnotám X a Y potom iba pričítame hodnoty 160 resp. 128. A teraz nasleduje výpis programu so stručným komentárom.

```
REM inicializacia obrazovky Low-Res, 32 farieb
SCREEN 1,320,256,5,1
```

```
REM Otvorenie okna
WINDOW 2,"Ukazka",,,1
```

```
R=100      :REM polomer kružnice
SX=160     :REM stred obrazovky v smere x
SY=128     :REM y-ova suradnica stredu obrazovky
PI=3.1415  :REM definovanie Ludolfovhovho cisla
```

```
FOR f=0 TO 2*PI STEP .01
  X=R*COS(f)+SX :REM vypocet x-ovej suradnice
  Y=R*SIN(f)+SY :REM vypocet y-ovej suradnice
  PSET (X,Y)    :REM vykreslenie bodu
NEXT
```

```
opakuje:      :REM cakaj na stlacenie
IF INKEY$="" THEN opakuje :REM lubovolnej klavesy
SCREEN CLOSE 1 :REM zatvor obrazovku
END
```

Ak ste program vyskúšali, môžeme pokračovať ďalej. V rodinnom príbuzenstve s kružnicou je elipsa. Ako ju nakresliť? Skúste zmeniť polomer kružnice R. Po spustení programu však dostaneme menšiu alebo väčšiu kružnicu. Ale akonáhle hodnota R pre výpočet X-ovej a Y-ovej súradnice bude rôzna, dostaneme elipsu. Skúste napríklad priamo dosadiť hodnoty 140 a 70:

$$X=140*\cos(f)+SX \quad Y=70*\sin(f)+SY$$

Teraz skúsme nakresliť špirálu. Po našich skúsenostiach kreslenia kružnice a elipsy sa už ľahko dovtipíme, že k tomu stačí plynule zväčšovať polomer kružnice. Najvhodnejšie je nájsť funkciu, ktorá je úmerná uhlu otočenia f. Napríklad:

$$R=Konst*f$$

Konst je číslo, ktorej hodnota závisí od počtu kružníc. Ak špirála bude zložená z desiatich kružníc, potom pre Konst odporúčam hodnotu 1,7. Program špirály bude vyzerať takto:

```
SCREEN 1,320,256,5,1
WINDOW 2,"Ukazka",,,1
```

```
R=0      :REM inicializacia polomeru
SX=160   :REM stred obrazovky v smere x
SY=128   :REM y-ova suradnica stredu
          :REM obrazovky
PI=3.1415 :REM definovanie Ludolfovhovho cisla
Const=1.7 :REM definovanie hodnoty narastu
          :REM polomeru
```

```
FOR f=0 TO 20*PI STEP .01
```

```
  X=R*COS(f)+SX
```

```
  Y=R*SIN(f)+SY
```

```
  PSET (X,Y)
```

```
  R=f*Const :REM výpočet nového polomeru
```

```
NEXT
```

```
opakuje:
```

```
IF INKEY$="" THEN opakuje
```

```
SCREEN CLOSE 1
```

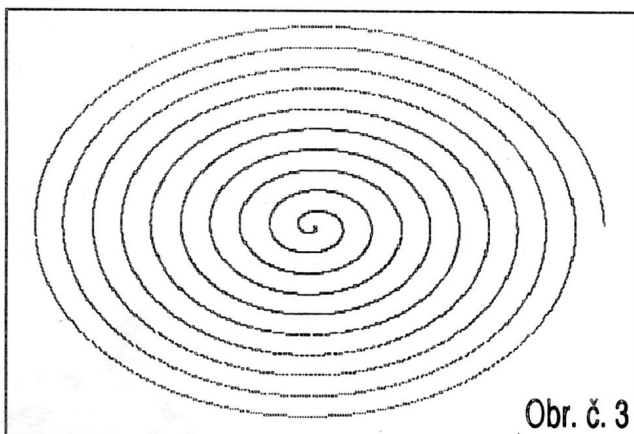
```
END
```

Mali by ste dostať obrázok podobný obrázku č.3. Nakreslená špirála sa nazýva Archimedesova špirála. Jej polomer je priamo úmerný uhlu otočenia. Ak na vyjadrenie vzťahu polomeru od uhlu otočenia použijete inú funkciu, dostanete logaritmickú špirálu. Skúste zodpovedajúce riadky zmeniť takto:

$$Const=0.05$$

$$R=Const*f*f$$

Polomer sa teraz mení kvadraticky v závislosti od uhlu otočenia. Môžete použiť aj inú funkciu,



Obr. č. 3

fantázii sa medze nekladú.

Ak už na zmenu polomeru nemáme viacej nápadov, skúsme niečo spraviť s funkciami sínus a kosínus. Z matematiky vieme, že funkciu kosínus môžeme nahradiť jej ekvivalentným zápisom:

$$\cos(f) = \sin(f + \pi/2)$$

A hneď sa ponúka myšlienka: posuv $\pi/2$ nahradiť inou hodnotou. V závislosti od posuvu dostaneme elipsy, ktorých os zvierá 45 stupňový uhol so súradnicovým systémom. Program bude nasledovný:

```
SCREEN 1,320,256,5,1
WINDOW 2,"Ukazka",,,1
```

```
R=100      :REM polomer
SX=160     :REM stred obrazovky v smere x
SY=128     :REM y-ova suradnica stredu
            :REM obrazovky
PI=3.1415  :REM definovanie Ludolfovho cisla
d=pi/2     :REM posuv
```

```
FOR f=0 TO 20*PI STEP .01
  X=R*SIN(f+d)+SX
  Y=R*SIN(f)+SY
  PSET (X,Y)
NEXT
```

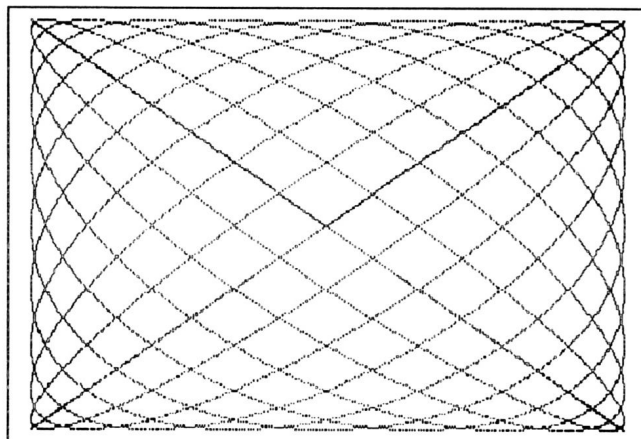
```
opakuj:
IF INKEY$="" THEN opakuj
SCREEN CLOSE 1
END
```

Ak $d=\pi/2$, dostaneme kružnicu; pre $d=0$ dostaneme úsečku a pre $d=\pi$ dostaneme tiež úsečku, ale kolmú na predchádzajúcu. Teraz skúsme nakresliť elipsy do jedného obrázku s

rôznymi hodnotami d . Už nebudem uvádzať celý program, iba jej hlavnú časť:

```
FOR d=0 to PI STEP PI/10
  FOR f=0 TO 2*PI STEP .01
    X=R*SIN(f+d)+SX
    Y=R*SIN(f)+SY
    PSET (X,Y)
  NEXT f
NEXT d
```

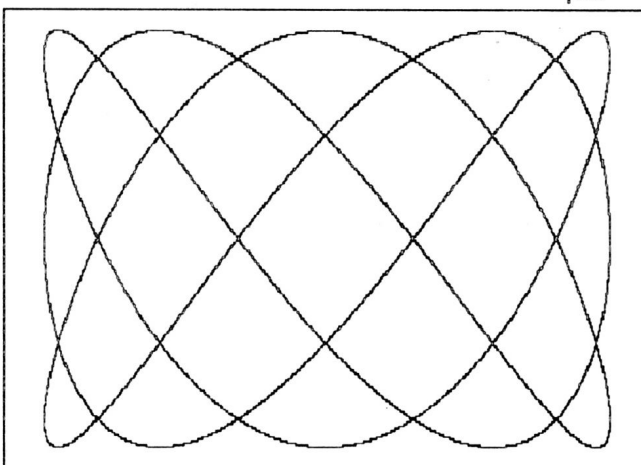
Na obrazovke dostaneme obrázok podobný nasledovnému obrázku.



Vráťme sa ešte späť k nášmu prvému programu - ku kružnici. Výrazy $\sin(f)$ a $\cos(f)$ skúsme zameniť za $\sin(f*a)$ a $\cos(f*b)$, kde "a" a "b" budú celé čísla. Ak spustíme program, dostaneme veľmi zaujímavé obrázky, ktoré sa nazývajú Lissajousove krivky (viď dole). Ak za a a b dosadíte malé celé čísla, experimentovaním môžete zistiť charakter kriviek.

Nabudúce si naše obrázky spestríme aj farbami. Zatiaľ príjemné experimentovanie.

-pm-





Imagine 5. část

HIERARCHIE A KOMPLEXNÍ OBJEKTY

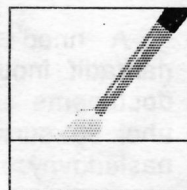
Pokud chcete udělat nějaký komplexní model, nebudete si asi přát vytvořit jediný monolytický objekt, reprezentující celé těleso. Když budete dělat les s 20-ti stromy, asi je nebudete chtít "vyřezat" z jednoho kusu. Je logické složit objekt z několika charakteristických částí. Pak si budete asi přát každé části přiřadit jinou barvu nebo materiál. Třeba okno - uděláte tabulky skla, dřevěný rámeček a tyto části spojíte. Dokonce i různé části jednoho objektu mohou mít rozdílné atributy.

K dispozici máte funkci, která vám spojí objekty dohromady. Když máte model, který chcete vyrobit (a uchovat!) z různých částí, dovolí vám Imagine ustanovit skupinu objektů, které budou pohromadě. Této skupině se říká grupa (group). Můžete s ní zacházet jako s celkem, ale můžete rovněž picknout libovolný objekt a manipulovat s ním nezávisle na ostatních.

Seskupení objektů je velice snadné. Pokud chcete "sgrupovat" objekty, clicknete myší na jeden, podržíte Shift, clicknete na druhý atd. Vzpomínáte, touto metodou můžete picknout více objektů najednou. Pak zvolíte funkci "Object/Group". Objeví se purpurová čára, spojující osy objektů. Objekt, který byl pickned jako první, bude "parent" (= rodič) skupiny. Z jeho os se rozbíhají čáry do os objektů nazývaných "child" (= dítě).

Rozdělení skupiny objektů zpět na jednotlivé

objekty je rovněž snadné. Stačí clicknout na parent objekt a tím se celá skupina stane picked. Pak už jen zvolíte příkaz "Object/Ungroup". Čáry spojující parent a child objekty zmizí a skupina je rozdělena.



Když vytvoříte skupinu, můžete s ní zacházet téměř identicky jako s neskupinovým objektem. Pokud chcete picknout celou skupinu, musíte clicknout na parent objekt. Můžete posunovat, zvětšovat, zmenšovat nebo rotovat skupinu objektů jako celek. Pokud však clicknete na child objekt, bude pickned pouze tento objekt, tedy ne celá skupina. S tímto objektem pak můžete dělat výše uvedené operace nezávisle na skupině nebo mu přidělit zcela odlišné atributy.

A aby toho všeho nebylo dost, můžete dokonce dělat skupiny skupin. Nebo skupiny skupin skupin. Nebo sku... ...zbytek si vnímavý čtenář jistě domyslí. Uděláte to přesně jako při spojování objektů. Držíte Shift a picknete první skupinu, druhou skupinu atd. Pak je spojíte pomocí "Object/Group". Někdy je výhodné takovéto "vnořené" úrovně skupin vytvářet, zvláště tehdy, když chcete později objekt animovat. Pěkným příkladem je třeba postava člověka. Můžete udělat tyto skupiny:

Prst - obsahuje všechny klouby

Ruka - obsahuje dlaň, 4 skupiny Prst a skupinu Palec

Paže - obsahuje skupinu Ruka, zápěstí, předloktí, loket

Tělo - obsahuje skupinu Hlava, trup, dvě skupiny Noha a dvě skupiny Paže

Tento druh spojování skupin se nazývá **HIERARCHIE**. Obrovská výhoda takové struktury je nasnadě - při animaci postavy. Picknete skupinu Paže a zrotujete ji kolem ramene. Všechny podskupiny paže tento pohyb následují, takže se paže pohne CELÁ! Nemusíte rotovat 14 kloubů, dlaň, palec, předloktí a tak dále. Pokud chcete trochu "doladit" polohu prstu, můžete manipulovat nezávisle s jeho klouby a paže jako celek zůstane nedotčená.

Někdy je užitečné dělat skupiny jednoduchých objektů, i když zrovna nepotřebujete vytvářet hierarchie. Pohyb objektů tím není omezen a navíc jim můžete snadno přiřazovat různé atributy.

Třeba když děláte lidskou tvář, můžete udělat oči jako skupinu vsazenou do hlavy místo toho, aby jste je zamodelovali přímo do tváře. Později můžete oči jednoduše změnit (změnit barvu nebo nahradit jiným typem (Oči z chromu! Cool!)) a nebo odstranit. Další velkou výhodou je možnost animace - osoba může kroutit očima a přitom třeba ještě měnit jejich barvu. Maximální efekt při minimální námaze.

PICK, ADD, DRAG. PICK, ADD, DRAG. TO JE NUDA!

Imagine umí samozřejmě ukládat a nahrávat objekty, které vytvoříte. K tomu slouží funkce **"Object/Save"** a **"Object/Load"**. O jejich významu jistě není třeba blíže hovořit. Imagine rozeznává různé typy objektů a je vhodné je při ukládání nějak označit. Název jako "Objekt1" vám toho druhý den asi moc neřekne. Proto se doporučuje doplnit za název některou z těchto koncovek:

- .iob** Imagine Object - lze nahrát do Detail editoru
- .iout** Imagine Object - faceless outline (obrys bez tváří)
- .ifm** Imagine Form - nahrává se do Form nebo Detail editoru
- .iff** - standardní obrázek nebo brush
- .ham** - obrázek nebo brush ve formátu HAM
- .iff24** - 24-bitový obrázek nebo brush. Nejvyšší kvalita
- .spth** Spline path - zakřivená zteзка
- .lpth** Line path - přímá zteзка

Dále je zajímavé zmínit se o ukládání objektů v souvislosti se skupinami. Když picknete parent

objekt skupiny, uloží se na disk skupina jako celek. Pokud picknete child objekt, bude uložen pouze on a ne skupina, do které náleží.

Dalším příkazem, který můžete aplikovat na picked objekt je **"Functions/Snap to Grid"**. Posune všechny picknuté objekty tak, že střed jejich os bude ležet v nejbližším uzlu mřížky. Velice užitečné pro precizní nastavení!

A pak tu máme několik užitečných příkazů: **"Object/Cut"**, **"Object/Copy"** a **"Object/Paste"**. **"Cut"** odstraní objekt ze "světa" Imagine a uchová ho v paměti. Když provedete **"Paste"**, objekt bude dosazen zpět na své místo, odkud byl odstraněn. Objekt je však stále v paměti a proto si můžete udělat jeho další kopie. **"Copy"** pracuje jako **"Cut"** ale s tou výjimkou, že objekt není po zkopírování paměti odstraněn ze scény. Poté můžete opět použít **"Paste"**.

Dalším důležitým příkazem je **"Functions/Join"** (spojit). Slouží ke stavbě komplexních objektů a spojí dva a více objektů do jednoho. Nový objekt bude používat osy prvně picknutého objektu. Je obtížné takto vytvořený objekt rozložit zpět. Proto spojte objekty u nichž předem s jistotou víte, že mají tvořit pevné těleso.

Příklad: děláte model auta. K bočním stranám připojíte zrcátka, střechu a podlahu. Vzpomeňte si na výhodu skupin. Kola asi nebudete chtít spojit. Naopak, pokud připojíte kola jako skupinu, můžete je později rotovat. Stejně tak můžete definovat chromové středy kol nezávisle na gumových pneumatikách. Takže skupiny jsou někdy výhodnější než spojené objekty, neboť jak je vidět, poskytují vám větší volnost.

Ve stejném menu se nachází funkce **"Merge"**. Je zajímavá tím, že odstraní všechny duplicitní faces, hrany a body ve vašem objektu. To je užitečné zvláště když spojujete objekty; některé jejich části se mohou shodovat a tím zbytečně zpomalovat zobrazování a výpočet. Velice jednoduchým příkladem je spojení dvou krychlí, které spojíte dvěma stěnami k sobě. Potom bude v místě spoje 8 hran a 8 bodů místo obvyklých čtyř. Funkce **"Merge"** však tento nedostatek odstraní.

"Functions/Delete" odstraní všechny picked objekty ze scény. Klávesnicový ekvivalent je Prává_Amiga-d.

Imagine má rovněž jednoúrovňové **Undo** (vrací o jeden krok zpět). Z klávesnice Prává_Amiga-u. Pokud provedete nějakou funkci a pak uděláte Undo, vrátí se situace zpět před provedení této funkce. Když použijete undo ještě jednou, objeví se opět účinky odstraněné funkce.

ATRIBUTY A BARVY

S níže popsanými nízkourovňovými příkazy už jste schopni sestavit objekt. Další úrovní tvorby objektu je definice barvy povrchu a atributů objektu. Rovina může být sestavena ze dvou trojúhelníků. Ale v závislosti na nastavení atributů může vypadat jako tabulka skla, zrcadlo, čtverečkovaný papír, deska dřevěného stolu nebo obraz vaší babičky. Imagine naštěstí poskytuje vynikající možnosti v definici charakteristik povrchu.

Každý objekt má sadu atributů, které můžete libovolně změnit. Ve skupině může mít každý objekt atributy nezávislé na parent objektu. Když picknete celou skupinu, měníte pouze parametry parent objektu. Atributy objektu změníte tak, že ho picknete a zvolíte **"Object/Attributes"**. Objeví se requester s bohatou nabídkou, kterou si ale podrobně vysvětlíme později. Rovněž je zajímavé, že v tomto requesteru kromě změny atributů objektu můžete změnit i jeho jméno.

MÓDY

Základní příkazy jsou velice důležité, neboť je používáte neustále. Ale k dispozici máte ještě další významnou pomůcku - různé módy, ve kterých můžete provádět různé operace. Některé módy vám dovolí pracovat se skupinami. Jiné vám umožní picknout a manipulovat nikoliv s objekty, ale s body, hranami a faces. S některými zase můžete přidat nové body, hrany a faces.

Módy můžete snadno změnit. Nachází se všechny v menu **"Mode"**. Aktivní mód je vždy zobrazen v horní liště. Klávesnicovými ekvivalenty jsou Prává Amiga a číslo, které je uvedené v "Mode" za názvem módu.

Implicitní mód je **"Pick groups"**. Znamená to, že cokoliv ve skupině picknete, bude celá picked. Pokud chcete picknout individuální objekt, dokonce i když je parent skupiny, zvolte mód **"Pick objects"**. Když nemáte objekty ve skupinách, chovají se "Pick groups" a "Pick objects" identicky.

Další módy vám dovolí zacházet s různými částmi objektu. Až do teď jste byli zvyklí zacházet s celým objektem najednou. Mohli jste ho

rotovat, zvětšovat, zmenšovat, přemísťovat, připojit ke skupině atd. ale to vše bez ovlivnění jeho základní struktury. Zbývající módy se tedy týkají ČÁSTÍ objektů, ne objektů jako celků. Důležitá poznámka - pokud chcete vstoupit do některého z těchto módů, musí být selected nejméně jeden objekt nebo skupina.

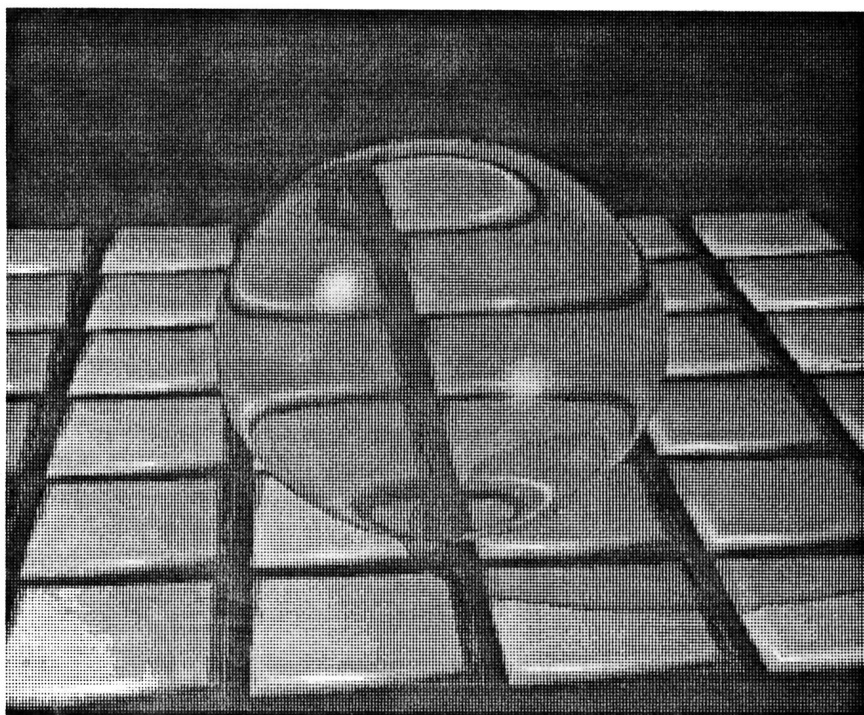
Všechny příkazy pro **Pick a Select** pracují stejně bez ohledu, zda se jedná o pick faces, edges a nebo points. Většina ostatních příkazů jako např. Delete bude pracovat i na libovolných částech objektů.

Nejdříve si vysvětlíme mód **"Pick points"**. Pokud picknete nějaký objekt nebo skupinu a vstoupíte do módu **"Pick points"**, objekt se zbarví bíle a už NENÍ picked. Jeho body budou viditelné (zobrazí se jako malé čtverečky). Uvědomte si, že už jste v jiném módu; už nepracujete s objekty jako dosud, ale s body. Když clicknete myší na nějaký bod, zbarví se do oranžova jako když picknete objekt. Můžete použít klávesu Shift a picknout více bodů najednou. Rovněž můžete použít jakoukoliv z pomůcek pro Select.

Selected body jsou zelené, picked body jsou oranžové a picked&selected body jsou žluté.

Příště dokončíme módy a uděláme si nějaké pěkné ray-tracingové obrázky.

Podle tutorial manuálu Stevena P. Worleye
volně píše: Jan Slanina



GFX-RIP



Popis pro verzi 1.2
(Programoval MR. Mad z
United Forces).

OVLÁDÁNÍ

Všechny funkce se ovládají levým tlačítkem myši. Rychlost nárůstu funkce můžeme zrychlit současným stlačením levého a pravého tlačítka myši. Například u funkce "Height":stiskneme-li levé tlačítko, hodnota se zvětší nebo zmenší o jedničku. Stikneme-li současně i pravé tlačítko, přičítání nebo odečítání se stále opakuje.

HORNÍ BLOK

Width:

Udává šířku obrazu, která se uloží. Modulo je přitom nezávislé (důležité u obrázků, u kterých jsou bitové mapy těsně vedle sebe).

Height:

Udává výšku bitových map. Při rychlém běhu se počítadlo zastaví na hodnotě 200 a 256, kde zůstane jednu sekundu stát.

Depth:

Počet bitových map.

Modulo:

Udává kolik hodnot se má přeskočit na každém řádku.

DDFStart:

Udává "DataFetchStart" z levé strany obrazovky.

DDFStop:

Udává "DataFetchStop" z pravé strany obrazovky.

BPL1-BPL6:

Toto jsou ukazatele bitových map. To ON/OFF udává ukazatel aktivní.

SPODNÍ BLOK

Color:

Přepíná mezi ukazovátkem bitové mapy "BPL-Pointer" a barvou.

SearchCopLst:

Prohledá barvy Copperlistu

SearchColTab:

Hledá v paměti tabulku barev. Pomocí pravé klávesy myši vyhledáváme tabulku barev. Kurzorovými klávesami je možné scrolovat paměť. SpaceBar přepíná mezi "ColMap" a "CopMap". V "ColMap" módu se vychází z toho, že barvy jsou v paměti uloženy těsně vedle sebe, zatím co v "CopMap" módu program vychází z toho, že mezi barvami stojí právě jedno prázdné slovo (např. u CopperListu).

HAM-HB:

Přepíná z HAM módu do HB módu a naopak.



Dec-Hex:

Přepíná mezi decimálními a hexadecimálními hodnotami.

Standard:

Nastaví vše do původního stavu.

Forward-BackWard:

Udává směr vyhledávání barev a bitových map(odpředu nebo odzadu).

Quit:

Opuštění programu. Můžeme kliknout do levého horního rohu.

BPLptr:

Po nakliknutí této funkce se ukáže kurzor v povelovém řádku, pomocí něhož můžeme nastavit počáteční (startadress) adresu pro vyhledávání Copperlistů. Kromě toho se všechny ukazatele, které nejsou nastaveny na OFF, nastaví na tuto adresu.

SearchCopLst:

Jestliže vyhledá CopperList v paměti, dosadí eventuálně DDFStart, DDFStop, Modulo a Hi-res nebo Lo-res.

SearchManual:

Prohledává se paměť ručně pomocí myši. Podléhají jen ty ukazatele, které jsou přepnuty na "ON".

Invert:

Invertuje ukazatele (ON-OFF).

Reset:

Smaže všechny ukazatele.

Clear:

Smaže pouze jeden ukazatel (podle výběru).

Swap:

Vymění dva ukazatele.

Copy:

Zkopíruje jeden ukazatel do některého jiného.

Move:

U obrázků, které namají kompletní rozlišení, může se pohnout viditelným výřezem.

Save:

IFF-ILBM:

Uloží se obrázek se zadanými hodnotami jako IFF-Pic (Není zpakovaný).

RawNorm:

Uloží obrázek jako bitové mapy těsně vedle sebe.

RawBlit:

Uloží obrázek řádek za řádkem. (1 řádek 1 mapa, 1 řádek 2 mapa)

CMAP:

Uloží jen barevnou mapu jako zdrojový kód pro Copper.

LoRes:

Přepíná rozlišení mezi Lo-res a Hi-res módem.

SearchChip-FastMem:

Toto nastavení umožňuje Riperu prohledávat FastMem, přičemž všechny funkce, které pracují s barvami a Copperlistem jsou bez omezení.

Verze 1.1:

Obrázky můžeme nyní hledat při nakliknutí "SearchManual" také kurzorovými klávesami, přičemž se může z těchto funkcí vystoupit pomocí klávesy ESC.

Chyby:

- SearchChip - Search FastMem
- Save Function
- Cli Function

Verze 1.2:

Klávesové funkce.

- F1 => Height na 200
- F2 => Height na 256
- F3 => Height na 400
- F4 => Height na 512
- FA => Quit
- H => Hi-Res
- L => Lo-Res
- ? => Modulo = 0
- > => Modulo +2
- < => Modulo -2
- M => Modulo +10
- N => Modulo -10
- [=> DDFStop -
-] => DDFStop +
- + => Depth +1
- => Depth -1
- 1-6 => BPL-ukazatel 1-6
- 7 => Invertuje BPL-ukazatel
- F => Uvolní všechny ukazatele
- SPC => BPL1+size=BPL2
- BPL2+Size=BPL3
- TAB => Uloží IFF
- ~ => Uloží RAW-NORM
- X => Hledání Copper-barvy
- C => Hledání barevné mapy
- Q => Hledání Copperlistu pro

BPL+Colors

W => Ruční hledání barevných map

R => Reset BPL-ukazatele

A => Move

S => Standart

D => Search Direction (od zadu)

B => Search-Pointer

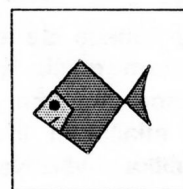
ESC => Konec funkce

V ručním vyhledávání:

R-ALT => Pointer = Pointer + Size

R-AMIGA => Pointer = Pointer - Size

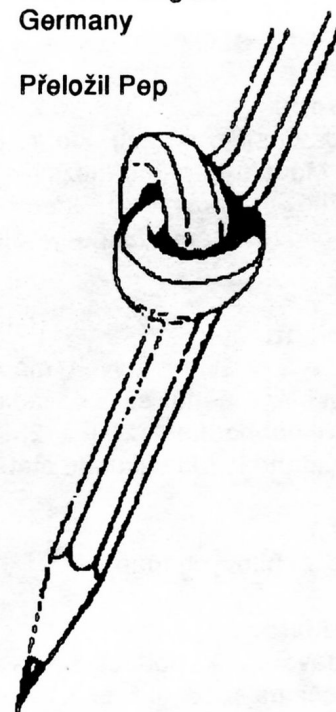
Tento program je Public Domain....



Případné vaše dotazy, chybová hlášení nebo tipy pište na tuto adresu:

F.Hofmann
Klare Perle 17
W3320 Salzgitter
Germany

Přeložil Pep



Dejiny Amigy ?

Viete ako sa zrodila AMIGA ? Chcete niečo vedieť aj o minulosti svojho miláčika ? Tak si prečítajte nasledujúce riadky.

Všetko sa začalo asi pred 8 rokmi. Americká firma Amiga vyvíjala nový hrací automat a hľadala sponzorov. Finančná situácia firmy nebola slávna a tak ponuka firmy Commodore prišla veľmi vhod. Commodore však nechcela iba hrací automat, ale aj plnohodnotný a kvalitný počítač. A dokonca taký, ktorý by "zatieňil" všetky ostatné.

Termíny zhotovenia nováčka boli relatívne krátke a vývojári stáli pred problémom. Počítač potrebuje predovšetkým užívateľskú platformu a DOS (diskový operačný systém). Jadro prevádzkového systému Amigy už bolo na svete: multitaskingové jadro "EXEC" poskytujúce široké možnosti, "graphics.library" a aj "Intuition.library".

Zabudlo sa však na jednu podstatnú vec: DOS. Firma Metacomco mala v tom čase funkčný DOS pre počítače operujúce s procesormi MC68000 - TRIPOS. Commodore sa chytil tejto záchranej slamky a kúpil ho. Historka vraví, že vývojár TRIPOS-u sa zatvoril s prototypom Amigy, výtláčkami dovtedy hotového operačného systému, a keď po týždni vyšiel, bol DOS hotový. A zlé jazyky tvrdia, že to je na DOS-e aj vidieť.

A naozaj: DOS sa v operačnom systéme Amigy správa trochu cudzo. Väčšia časť operačného systému (OS) je písaná v jazyku C alebo assembleri a Tripos a teda i Amiga DOS je vyvinutý v jazyku BCPL. Hoci C a BCPL majú spoločného prarodiča (B), BCPL má iný systém prístupu do pamäti. Reťazce znakov nie sú, ako káže konvencia, ukončené nulovým bytom, ale začínajú bytom určujúcim ich dĺžku.

Takto chaoticky bol vytvorený operačný systém AMIGY 1.0. Vývojárom behal mráz po chrbte, keď videli tú kopu chýb a neprehľadností, ale z časových dôvodov to inak nešlo. Čo teraz? Vypustiť do predaja Amigu s nedokončeným operačným systémom, alebo odložiť premiéru počítača? Tu prišla spásna myšlienka: prečo nahrávať operačný systém do ROM? Dajme ho na disketu a môžeme ho pohodlne ďalej vyvíjať. Tak bolo ku "gigantickej" hlavnej pamäti Amigy 1000 (256 KB) pridaných ešte ďalších 256 KB RAM-ky, do ktorej sa pri štarte systému nahráva jadro operačného systému. Disketa, na ktorej sa nachádzal operačný systém, sa nazvala "KICKSTART". Amiga bola slávnostne uvedená na trh a operačný systém sa mohol ďalej vyvíjať. Z verzie 1.0 vzniká 1.1 - je síce lepšia, ale ešte stále nie tá pravá.

Amiga OS 1.1 sa rozšíril o ďalšie potrebné nástroje a očistil sa skoro od všetkých chýb. Výsledok nesie číslo 1.2 a slúžil skoro 5 rokov ako základ pre všetky AMIGA počítače. V Amige 500 a Amige 2000 bol operačný systém umiestnený do pamäte ROM, čo užívatelia Amigy 1000 privítali so škodoradosnými úsmevmi, lebo si mysleli, že budúcnosť patrí nahrávateľnému Kickstartu (čo sa ukázalo ako kolosálny omyl: OS 2.0 má 512 KB a nedá sa nahráť do "kickstartovej" RAM-ky Amigy 1000).

Potom klesli ceny pamäti a pevné disky sa stali prístupnými i pre taký počítač ako je Amiga. Bohužiaľ, vývojári Amigy zabudli na jedno: že Amiga DOS bol štartovateľný iba z mechaniky "DF0". Bez diskety v DF0 sa nič nedialo, či už s pevným diskom alebo bez. Preto mierne upravili Kickstart tak, aby sa systém dal bootovať aj z harddisku. Ale to je už vlastne ďalšia verzia Kickstartu označovaná 1.3. Avšak zmeny v Kickstarte oproti verzii 1.2 sú minimálne. Podstatné zmeny boli uskutočnené na diskete Workbench, kde pribudol "Shell" a nové príkazy do adresára "c".

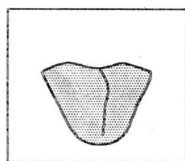
Takže Kickstart 1.3 bol na trhu. Onedlho sa začalo hovoriť o novej verzii 1.4, ktorá má byť lepšia, krajšia a úplne bezchybná. Ale Commodore si už nemohla dovoliť uviesť na trh ďalšiu nedokončenú verziu systému. Potajomky sa v dielňach pracovalo na novej verzii, o ktorej nikto nič nevedel. Vývojári dokonca urobili menší žartík: zobrali verziu Kickstart 1.3, zmenili jej okná a pod označením 1.4 - Alfa ju pustili do obehu. A Amigisti naleteli...

Trvalo 3 roky, kým nový Kickstart, nazvaný 2.0 predstavili verejnosti: súčasne s ním sa objavila i Amiga 3000. Avšak A 3000 mala vstavanú iba beta-verziu Kickstartu. Trvalo ďalší dlhý rok, kým OS 2.0 prišiel na trh, ale vyplatilo sa: OS 2.0 plní všetky očakávania užívateľov i programátorov a je považovaný za najzrelejší operačný systém osobných počítačov.

A to najlepšie nakoniec: kým menej komfortné operačné systémy pre počítače s procesormi Intel, ako je WINDOWS alebo OS/2, vyzdvihujú svoje "skromné" požiadavky ako je napr. minimálne 16 MHz, 4 MByte RAM, rýchly pevný disk, OS 2.0 beží aj na dobrej starej A1000 so 7 MHz a 512 KBytmi RAM ešte rýchlejšie!

Kto už raz prišiel do styku so Shell pod OS 2.0, už sa s ním tak rýchlo nerozlúči. A podľa toho, čo sa rozpráva - vývoj v Commodore nestojí - nechajme sa teda prekvapiť.

Ing. Miroslav Pavlík



ARexx

ing. Miroslav Pavlík

ARexx - pojem, s ktorým sa v poslednej dobe často stretávame, ktorému však naše odborné časopisy nevenujú žiadnu pozornosť.

ARexx je súčasťou diskety Workbench 2.0 a dodáva sa k počítačom Amiga pracujúcim pod Kickstartom 2.x. ARexx sa však neviaže na nový systém, rovnako spoľahlivo funguje aj na starej Amige 1000.

Najprv by sme mali zodpovedať otázku - čo je to ARexx ? Ak si prečítame niektoré "definície", dospejeme k názoru, že je to programovací jazyk podobný Basicu, ale skôr scriptový jazyk, alebo je to nástroj na komunikáciu medzi rozličnými programami ? Treba povedať, že v každej charakteristike je časť pravdy.

Jadro balíčka tvorí interpreter. Interpreter je program, ktorý prekladá program ARexxu do formy zrozumiteľnej procesoru - do strojového kódu. Na rozdiel od kompilátora, ktorý prekladá program tiež do strojového jazyka, ale uchováva ho v pamäti v binárnej forme, ktorý sa potom vykonáva, interpreter to robí počas práce počítača. Takéto počínanie má ako prednosti tak aj nedostatky, o ktorých sa však nechcem rozširovať. Iba pripomínam, že ARexx interpretuje program.

Interpreter je odštartovaný volaním programu "RexxMast", ktorý sa nachádza v adresári "System" diskety Workbench 2.0. Program napísaný v ARexxe sa spustí až príkazom "rx", ktorého argumentom je názov programu zakončený príponou ".rexx". Všetky programy štartované pomocou "rx" očakáva interpreter v logickom adresári "Rexxc:", ak neuvedíme inú prístupovú cestu.

Na ukážku, ako pracuje ARexx, si napíšme krátky program. Najprv však niekoľko slov o

inštalácii ARexxu.

Ak váš počítač pracuje pod systémom 2.0, stačí odštartovať disketu Workbench 2.0 a v adresári System dvakrát nakliknúť ikonu RexxMast. Ak pracujete pod systémom 1.x, potom na diskete, z ktorej bootujete počítač, vytvorte adresár "Rexxc", do ktorého nakopírujte obsah rovnomenného adresára z diskety WB 2.0. Do adresára "c" umiestnite program "RexxMast" a do "libs" súbory "rexksupport.linbrary" a "rexksyslib.library". Potom otvorte Shell alebo CLI a odštartujte program RexxMast vpísaním jeho názvu.

Zdrojový program môžete napísať ľubovoľným textovým editorom. Náš prvý program nech vyzerá takto :

```
/* Prvy program */
say "Ema ma mamu"
```

Vytvorený textový súbor uložte do RAM: pod názvom "Ema.rexx" a odštartujte ho nakoniec zo Shellu príkazom "rx ram:ema".

"Say" odpovedá príkazu PRINT v Basicu, alebo "ECHO" z Amiga DOS a vypíše reťazec znakov na obrazovku. Znaky "/" a "/" označujú komentár, to znamená, že všetko, čo je medzi nimi, je interpretom ignorované a nemá vplyv na program. **Každý ARexx program musí začínať komentárom!**

Samotný jazyk vyvinul v polovici osemdesiatych rokov Američan M.F. COWLISHAW a v určitých súvislostiach sa skutočne blíži BASICu. Popri ľahkej zvládnuteľnosti, ktorá je vlastná obom jazykom, je najzreteľnejšou spoločnou vlastnosťou komfortné spracovanie reťazcov. ARexx tu ide dokonca o niečo ďalej ako Basic, zásoba stringových funkcií je takmer nevyčerpatelná.

Ďalšou výhodou je, že nemusíme deklarovať premenné. Jazyky ako C alebo OBERON požadujú

od programátorov deklaráciu premenných. Je to zápis, ktorý oznámi kompilátoru, ktoré premenné budú použité a predovšetkým, čo budú obsahovať. Rozlišuje sa tu napr.

medzi číslami, jednotlivými znakmi a reťazcami znakov. ARexx nepozná tieto rozdiely, všetky premenné sú rovnakého typu. Obsah premennej môže byť priradený hociktovej inej bez obmedzení. To však prináša so sebou problémy, s ktorými sa interpret musí vyrovnávať. Tak napríklad, každá funkcia, ktorá prevezme premennú ako parameter, musí najprv vyskúšať, či s obsahom premennej dokáže čosi urobiť. Pre matematickú funkciu to znamená, že sa musí presvedčiť, či premenná obsahuje číslo a nie reťazec znakov.

Čo na prvý pohľad vyzerá ako nedostatok (programy sú pomalšie), stáva sa pri bližšom pohľade prednosťou. Reťazce (stringy) môžu obsahovať všetky známe znaky v ľubovoľnom poradí. Výsledok výpočtov sa dá bez problémov prekonvertovať ktoroukoľvek stringovou funkciou, napr. preformátovať na výstup do tabuľky. Hoci ARexx nebol vyvinutý ako programovací jazyk, nato je príliš pomalý, predsa len má viac výhod ako nevýhod.

Základné príkazy ARexxu môžeme ďalej rozšíriť externými príkazmi z knižníc. Takto má programátor k dispozícii špeciálne príkazy, ktoré môže používať vo vlastnom programe. Uvedená koncepcia približne zodpovedá Shared - Libraries, ktoré sú zvyčajne v adresári "LIBS".

Ďalšie, značne širšie, možnosti predstavujú ARexxové porty. Použitím portov dostáva bežiaci ARexxový program charakter "listovej schránky", v ktorej sa môžu uschovávať ľubovoľné správy. Tieto správy obsahujú príkazy, ktorým užívateľský program rozumie a spracúva ich. Užívateľský program potom môže opačne odovzdávať výsledky pre ARexx. Každá aplikácia ARexxového portu pozná určitý počet základných príkazov. Každý programátor užívateľských programov, ktorý sa drží smerníc a implementuje zaužívané príkazy, uľahčuje užívateľovi vstup do svojho produktu. V podstate môžeme povedať, že dobrý ARexxový port pozná všetky funkcie, ktoré má program obsiahnuté v hlavnom menu a v oknách.

ARexxový port oslovujeme príkazom "address", po ktorom nasleduje názov portu prípadne ďalší príkaz. Ak na konci chýba príkaz, znamená to, že zvolený port je aktuálny pre všetky nasledujúce. Teda všetky príkazy sú zasielané na tento port, až kým nepôjde o interný príkaz.

Aktuálny port môže byť oslovený iba príkazom, ktorý nepatrí k základným inštrukciám jazyka ARexx, a ktorý sa nachádza v zozname externých príkazov. Ak si užívateľský program nevie poradiť s príkazom, zaslaným na oslovený port, vyšle hlásenie o chybe a preruší prácu s ARexxovým scriptom.

Existujú aj iné programy podobné ARexxu, ktoré však majú niektoré nevýhody. Pre programátora nie je výhodné implementovať ARexxový port namiesto kompletného scriptového jazyka, hlavne pri lacnejších programoch. K tomu sa pripája aj tá skutočnosť, že ARexx je už značne rozšírený a do určitej miery štandardný medzi scriptovými jazykmi. Užívateľ sa nemusí učiť nový jazyk pre každý program. Dajú sa využiť staré poznatky aj pri tvorbe nových vecí. Stačí ich doplniť o špecifické príkazy.

Pre lepšie pochopenie možností ARexxu vám načrtnem jeden konkrétny príklad využitia.

Najväčšou výhodou ARexxového programu je možnosť oslovovať viac portov rozličného použitia za sebou. Aj tu sa využívajú možnosti multitaskingu Amigy. To má za následok, že viaceré programy, ktoré bežia súčasne v pamäti, sa vzájomne riadia a synchronizujú, a nakoniec si vymieňajú potrebné dáta. V praxi tak môžeme zostaviť script, ktorý pomôže vytvoriť sériové dopisy za pomoci textového editora a databázového programu. Script môžeme odštartovať z editora, pričom predpokladáme, že list je už v editori. V databánke ARexxový program automaticky otvorí kartotéku, v ktorej sa vyhladá zodpovedajúca veta podľa požiadaviek. Teraz vstupuje do hry textový editor, ktorý prečíta nájdené dáta a umiestni ich do listu. V ďalšom sa aktivuje program Point, ktorý odošle list cez elektronickú poštu (modem, fax), čo je spravidla rýchle a lacné. Proces sa opakuje, pokiaľ sa vyčerpajú všetky adresy.

Takéto a iné možnosti vám poskytne ARexx. Čas ukázal, že žiadny program sa dnes bez ARexxových portov nezaobíde.

Takže jedno si treba zapamätať: ARexx nie je náhradou za jazyky ako C alebo Modula či Oberon a ani nikdy nebude. Je to nástroj na komunikáciu viacerých programov. V ďalších číslach AMIGA staru prinesieme popis základných príkazov a funkcií jazyka.



PowerPacker & ARexx

Ing. Miroslav Pavlík

Kto často používa program PowerPacker, určite vie, že zkomprimovanie krátkych súborov je efektívnejšie pomocou nastavenia "Very Good" alebo "Good", ako Best - dekompresia potom trvá kratšie.

Príčiny tejto skutočnosti treba hľadať v programe, vo svojom príspevku sa však nebudem rozširovať o tejto problematike. Pokúsme sa však nájsť riešenie problému.

Ak teda chceme nájsť najefektívnejšie riešenie, musíme otestovať všetky 3 varianty a nakoniec si ponechať najkratší výsledok. Manuálne uskutočniť uvedený algoritmus nie je zložité, ale oveľa jednoduchšie a pohodlnejšie je naprogramovať ho. Ako stvorený na takéto aplikácie je jazyk ARexx, s ktorým už spolupracuje nová verzia PowerPackeru - 4.0.

Program prepíšete do ľubovoľného textového editora a uložte pod názvom "PP.rexx" do adresára "Rexx". Spustíte program RexxMast a potom PowerPacker. Z menu si zvolíte položku "ARexx Script..." a vpišete názov programu - "PP.rexx". Hneď sa otvorí requester "Load File" a môžete si vyvoliť spracovaný súbor. Po potvrdení requestera sa dáta 3x zkomprimujú a výsledok sa uloží do adresára "T:" (v RAM). Názov súborov sa vytvorí zo starého mena a z aktuálneho módu. Napríklad súbor "TestFile" mutuje na "T:TestFile.Best" alebo "T:TestFile.Good" alebo "T:TestFile.VeryGood". Potom sa ďalší requester spýta, či má pôvodný súbor s najkratšou variantou prepísať alebo nie. Nakoniec sa 3 pokusné súbory vymažú.

Ako pracuje ARexx-program, sa dozviete z návodov, prípadne sledujte kurz ARexxu v našom časopise.

Ako vidíme, ARexx sa dá skutočne využiť ako riadiaci software a verím, že sa mi podarilo vo vás vzbudiť záujem o tento jedinečný programovací jazyk.

Použitá literatúra:

AMIGA 3000 Handbuch zur Systemsoftware
Amiga DOS, 1992, číslo 3

```
/* ARexx-script pre PowerPacker */
/* Najde najefektívnesi spôsob komprimovania */
```

```
options results
address PowerPacker
```

```
file="req"
name="dummy"
BestLen=Try("Best")
VeryGoodLen=Try("VeryGood")
GoodLen=Try("Good")
```

```
vitaz = max(BestLen, VeryGoodLen, GoodLen)
```

```
if vitaz=BestLen then
  NajMod=Best
```

```
else
  if vitaz=VeryGoodLen then
    NajMod=VeryGood
  else
    if vitaz=GoodLen then
      NajMod=Good
```

```
EZRequest "Najlepsí výsledok sa dosiahol pri "NajMod".
Uložiť ?"
```

```
address command
```

```
if rc=5 then
  copy "t:"name"."NajMod"."file"
```

```
delete ">NIL: t:"name".best"
delete ">NIL: t:"name".good"
delete ">NIL: t:"name".verygood"
```

```
exit
```

```
Try: procedure expose file name
```

```
parse arg mode
interpret "eff" mode
```

```
if file="req" then
  load
else
  load file
```

```
zisk=SUBWORD(result,2,1)
```

```
if file="req" then do
  GetFileName
  name=COMPRESS(result)
  GetFullName
  file=result
end
```

```
save "t:"name"."mode"
return zisk
```





Počítačová hudba alebo



MIDI

MIDI je skratka anglických slov "Musical Instrument Digital Interface", ktoré môžeme voľne preložiť ako digitálne spojenie hudobných nástrojov. Prostredníctvom tohoto normovaného systému je možné prepojiť rôzne hudobné nástroje medzi sebou, ako aj počítačom.

Pozrime sa najprv na "dejiny" MIDI. S rozšírením syntetizérov koncom 70. rokov bola zrejmá potreba prepojiť jednotlivé hudobné nástroje. Boli síce vypracované koncepcie umožňujúce prepojiť nástroje toho istého výrobcu, ale systém, ktorý by umožnil prepojenie nástrojov rôznych výrobcov, neexistoval. Úlohou bolo vypracovať systém, ktorý by bol jednoduchý, technicky ľahko zrealizovateľný a v neposlednom rade lacný. Predstavitelia mnohých firiem ako Roland, Yamaha, Korg, Kawai sa zišli roku 1982 a prijali systém navrhnutý firmami Roland a Oberheim a dali mu názov MIDI. Pretože štandard bol v určitých detailoch nejasný, výrobcovia si ho interpretovali po svojom a zrodili sa výrobky, ktoré neboli úplne kompatibilné. Preto boli založené normatívne orgány zodpovedné za dodržiavanie MIDI normy a jej ďalší vývoj. Konečne roku 1985 vyšla podrobná MIDI norma a odvtedy sú s kompatibilitou minimálne problémy. Systém je však neustále vo vývoji a jej nové rozšírenia sú zakotvené v dodatkoch (ako napr. kontroléry, Standart MIDI File Format, MIDI File Dump, MIDI Time Code, MIDI Machine control a iné). Doslova denne vznikajú nové dodatky a aplikácie. MIDI sa stalo priemyselným štandardom a žiadny výrobca hudobnej elektroniky si ho nemôže dovoliť ignorovať. Pôvodný zámer MIDI - prepojenie syntetizátorov - sa natolko zdokonalil a ďalej rozvinul, že dnes nám už umožňuje aj komunikáciu so samplermi, mixážnymi pultami, videotechnikou.

Technické zázemie MIDI nie je tak komplikované, ako by sa na prvý pohľad mohlo zdať. Každé zariadenie, ktoré chce komunikovať s okolitým svetom za pomoci MIDI systému, musí byť na to konštrukčne vhodne upravené. Musí disponovať s prijímačom a vysielačom MIDI informácií. Tie sa skladajú z obvodov UART (univerzálny asynchrónny prijímač - vysielač), väčšinou sa používa výrobok firmy Motorola typ 6850, ďalej z bloku optočlenov (na oddelenie rušivých napätí), z bloku elektroniky a z vlastného vodiča, zakončeného päťkolíkovým konektorom DIN. Vodiče sú napojené na vývody 4 a 5, 2 je kostra. Vývody 1 a 3 sú nezapojené. Konektory by mali byť označené MIDI IN - na vstup informácií a MIDI OUT - výstup informácií. Môže byť pridaný aj konektor s označením MIDI THRU, na ktorom je priama kópia dát, prichádzajúcich do MIDI IN. Prenos dát sa deje sériovo v norme RS 232 s upravenou prenosovou rýchlosťou. Môže sa uskutočniť taktiež paralelne pomocou CENTRONICS. Tento prenos je síce rýchly, ale je limitovaný dĺžkou vodiča do troch metrov. V prevažnej väčšine sa používa sériový prenos. Uskutočňuje sa pomocou prúdovej slučky, kde je však prúd znížený na 5 mA a má negatívnu logiku. Prenos prebieha asynchrónne: 1. bit je štartovací, potom nasleduje 8 datových bitov a celú prenášanú slabiku ukončuje stop bit. Oproti štandardnému RS 232 je zmenená rýchlosť z 19.2 kBd na 31.25 kBd. Jadro MIDI systému - integrovaný obvod UART-umožňuje konvertovať dáta paralelnej formy do sériovej a naopak - a tieto dáta potom vysielať alebo prijímať. Pracuje s formátom jedného byte (t.j. 8 bitov). Kábel na prepojenie MIDI zariadení má byť dvojžilový, s tienením pripojeným na kontakt č.2 a môže byť maximálne 15 metrov dlhý.



Podrobný technický popis máme za sebou, teraz sa venujme praktickým záležitostiam.

Prvé nástroje, do ktorých sa montovalo MIDI, boli syntetizátory. Postupom času sa MIDI objavilo aj v gitare, bicích nástrojoch, saxofóne a ďalších. Ako sa tvorí MIDI signál? Asi pomerne ľahko si to vieme predstaviť u syntetizátorov - stlačí sa klávesa, kontakt sa uzavrie a tým sa vyrobí impulz. Ale ako môžu vznikať MIDI dáta v gitare alebo saxofóne? Gitarový snímač, regulátor šumu a vzduchové senzory (u saxofónu) konvertujú el. impulzy a tlak vzduchu na elektrické napätie, ktoré sa ďalej mení na digitálne MIDI signály. Samozrejme takéto nástroje môžu vysielat iba MIDI-dáta.

MIDI zariadenia môžu dáta prijímať alebo vysielat. Predstavme si, že máme 2 syntetizátory. Spojme MIDI OUT nástroja A so vstupom nástroja B. Takto môžeme z klávesnice nástroja ovládať nástroj B. Podľa stláčaných kláves nástroj A generuje zodpovedajúce MIDI signály, ktoré prijíma nástroj B a vykonáva ich (hrá tóny). Nástroj A je riadiaci a označuje sa ako MASTER, nástroj B je riadený a nazýva sa SLAVE (otrok). Ak má byť "otrok" rovnocenný so svojím "pánom", musí sa spojiť výstup nástroja B so vstupom A. Oba nástroje môžu teraz prijímať i vysielat tóny hrané ako na A, tak aj na B.

Ak má byť vzájomne spojených viacero nástrojov, nástroje musia mať ďalší vývod - MIDI THRU, alebo potrebujeme MIDI-THRUBOX. Princíp oboch je rovnaký a ľahko zrozumiteľný. U MIDI nástrojov s konektorom THRU sa vstupné MIDI dáta nevyužívajú len na riadenie vnútorného generátora tónov, ale súčasne sa prenášajú ďalej v nezmenenej forme na konektor THRU. Nástroj má potom dva výstupy: OUT a THRU, ktoré sa však líšia. Na výstupe OUT sú len dáta generované nástrojom, kým na THRU sa nachádza presná kópia prichádzajúcich dát. Je to rozdvojka MIDI signálu. V praxi sa takto dajú za sebou zapájať viaceré nástroje (max. 3,4). MIDI THRUBOX obsahuje jeden MIDI IN vstup a viacero MIDI OUT výstupov. Použitím viacerých Boxov je možné prepojiť väčší počet nástrojov.

Doteraz bola reč iba o tom, že MIDI nástroj dokáže vysielat a prijímať dáta, ktoré sa vzťahujú na výšku tónu a jeho dĺžku. To je však len časť možností, ktoré MIDI-systém poskytuje. Normovaný MIDI-formát dovoľuje prenášať súčasne 16 celkom nezávislých kanálov. Ako porovnanie nám môže poslúžiť kábelová televízia, v ktorej tiež na mnohých kanáloch bežia rozličné programy k mnohým divákovi a to cez 1 kábel. Ak teda MIDI klávesnica vysielá dáta na kanále č. 3, tak prijímací kanál MIDI

nástroja musí byť tiež nastavený na 3. Preto skoro všetky MIDI prístroje obsahujú kanálový volič pre príjem i vysielanie. Niektoré prístroje s obmedzenou funkciou však vysielajú a prijímajú len na jednom kanáli (kanál 1).

Prostredníctvom jedného kanálu sa dá prenášať množstvo MIDI informácií. Týkajú sa tónu (ktorý kláves manuálu a kedy bol stisnutý a pustený), výmeny programov (ak sa zmení zvuk u riadiaceho nástroja, zmení sa i u "otroka") a riadenia (dáta pre vibračné efekty, výdrž, Portamento, úderová dynamika, zmeny výšky tónov). Cez kanály sa tiež môže prenášať signál taktu a tak môžu byť MIDI zariadenia synchronizované. Ak nástroj nevie jednotlivé dáta spracovať, lebo nemá zodpovedajúce funkcie, sú takéto dáta ignorované. MIDI systém ďalej pozná celý rad ďalších módo, systémových hlásení a synchronizačných príkazov, ktoré sú také rozsiahle, že by sa o nich dali napísať celé knihy. My sa však uspokojíme so základným princípom.

Takže systém MIDI kanálov umožňuje pri použití 16 kanálov súčasný prenos rozličných tónových a riadiacich dát do 16 nástrojov cez 1 kábel. Pritom vychádzame z toho, že jeden nástroj používa iba 1 kanál. To však nebýva vždy, lebo moderné MIDI-nástroje a zariadenia môžu prijímať informácie na viacerých kanáloch.

Niektoré nástroje obsahujú rytmické a doprovodné automaty s mnohými funkciami, ktoré generujú zvuk celej kapely. Ku kompletnému bicieho doprovodu sú generované ešte ďalšie doprovodné linky: basa, akordový vzor a prídavné sólové melodické sekvencie. Tieto zložky generovanej hudby sú nástrojom vysielané na oddelených výstupných kanáloch.

Pridelenie kanálov môže byť napríklad takéto:

Kanál	funkcia
1	= melódia (klávesnica-pravá ruka)
2	= basová linka (sprievodná automatika)
3	= akordový vzor 1 (automatika)
4	= akordový vzor 2 (automatika)
5	= sólové sekvencie (automatika)
10	= rytmický kanál (bicia automatika)

MIDI dáta sú vysielané na šiestich kanáloch, pričom kanály 2-5 a 10 sú generované automaticky nástrojom. Interný generátor taktu dokáže vysielat aj taktové impulzy cez MIDI tak, že externé MIDI-prístroje v návaznosti na klávesnicu sa zosynchronizujú. Výhody týchto postupov sú očividné. Jednoduchým spôsobom za podpory automatov sa dajú cez MIDI kanály riadiť ľubovoľné MIDI-nástroje, takže sa dá vytvoriť celý elektronický



orchester riadený z jednej klávesnice.

Pre profesionálov, ktorí nechcú použiť sprievodnú automatiku, poskytuje klávesnica prídavnú možnosť rozdelenia manuálu. Pritom je klávesnica rozdelená do 2 alebo viacerých častí, ku ktorým sú priradené rozličné MIDI-kanály.

Syntetizátor môže samozrejme naraz vyslať MIDI informácie, ale aj prijímať. Môže napríklad prijímať (a teda hrať) informácie pre basovú linku akordy, melodické zvuky, bicie paterny a pritom vyslať informácie o našej hre na klaviatúre. V spojení s Hard- alebo Softwarovými sequencermi (prístroj alebo počítačové programy na záznam a prehrávanie MIDI dát) majú multikanálové syntetizátory široké možnosti využitia.

MIDI dáta sú digitálne informácie, ktoré dokáže prijímať každý počítač. O korektnú komunikáciu sa stará MIDI Interface, ktorý sa u Amigy pripája na sériový port. Obsahuje MIDI-IN a podľa prevedenia jednu alebo viac MIDI-OUT a jeden MIDI-THRU konektor. MIDI software potom umožňuje bezprostrednú spoluprácu nástrojov popri prípade zariadení s počítačom. U MIDI programov sa prevažne jedná o sekvenčné programy (odhliadnuc od editorovacích pre programovateľné syntetizátory).

Softwarový sequencer má veľký počet stôp, na ktoré sa zaznamenávajú MIDI dáta alebo z ktorých sa znovu čítajú. Principiálne ich delíme na 2 skupiny. Prvá skupina dovoľuje záznam iba na jednu ľubovoľnú stopu, pričom ostatné stopy môžu byť zapnuté v prehrávacom móde. Prakticky to znamená, že sa potrebné stopy nahrávajú jedna za

druhou. Najčastejšie na najprv zaznamenáva bicia linka, potom basa a sprievodné akordy. Podľa rovnakých postupov potom nasleduje nahrávka melódie alebo sólových vstupov.

Tento postup praktizovaný profesionálmi je pre začiatočníkov nevhodný. Svoju tvorbu interpretujú najčastejšie iba jedným prstom za pomoci sprievodnej automatiky.

Preto zvlášť výhodné pre začiatočníkov sú softwarové sequencery druhej skupiny. Tieto umožňujú súčasnú nahrávku MIDI dát do viacerých stôp. Počas hrania nástroj generuje MIDI dáta automatických doprovodov na rozličných kanáloch a program ich zaznamenáva do oddelených sekvenčných stôp. A nakoniec sa dá samozrejme každá stopa editovať rôznorodým spôsobom. Chyby môžeme korigovať, eventuálne dopĺňať chýbajúce časti. Dajú sa tiež nahráť na ďalšie stopy metódou jedného kanála, napr. vyrobiť sólové hlasy pre ďalší MIDI nástroj.

V prehrávacom móde sequencery oboch skupín vysielajú MIDI dáta na zvolených kanáloch, takže MULTI-kanálový syntetizátor prehráva a vykonáva MIDI-príkazy.

Nabudúce vám predstavím MIDI sekvenčery pre Amigu, ktoré sa líšia cenou i ponukou funkcií.

-pm-

Použitá a odporúčaná literatúra:

Horký K.: Diplomová práca. Brno JAMU, 1986

GURU, 1991, číslo 2

Elektrón, 1986, číslo 12, strana 42-45

AMIGA Special, 1991, č. 10, str. 25-36

International MIDI Association: Dokument MIDI 1.0

INZERCIA

PD software.

Slavomír Jasenec,
Krahovská 13, 04012
Košice

Podmienky inzercie:

Uverejnení privátného inzerátu je bezplatné. Ceny plošné inzercie je včetně grafické úpravy 10.- Kčs za cm2. Poskytujeme slevy na plošnou reklamu podľa individuálnej domluvy.

Redakce si vyhradzuje právo neuverejnit inzeráty škodící jménu časopisu. Inzeráty budou zveřejňovány tak, jak jsou doručeny na adresu naší redakce. Inzeráty nesmějí porušovat Zákon o autorských právech.

Súrne zháňam program PowerPacker od verzie 3.13 vyššie a Imploder v4.0. Ďalej mám záujem o

Zháňam informácie o VIDEO BACKUP systéme na videorekordér pre počítače AMIGA alebo kontakt na majiteľov tohoto systému.

Ján Falatko, Kótayho 19, 04001
Košice, tel.: 095/33177

Predám prídavnú pamäť 512 KB s hodinami a prepínačom. Cena 1700 Kčs.

M. Škunda, nám. M. Slov. 18/74,
96501 Žiar, tel.: 0857/3041

Výhodne predám počítač Commodore AMIGA 500 (1 MB RAM, myš, joystick, TV-modulátor,

literatúra, 160 nahratých disket). Cena dohodou.

O. Lupták, Gorkého 2330/2,
96001 Zvolen

Pro AMIGU prodám externí disketové mechaniky 3.5" - 880 KB (2900), 5.25" - 360KB, vhodná pro emulátor a přenos dat počítačů PC (1400).

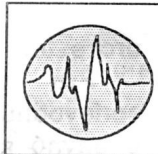
L. Nemec, M. Bureše 818, 57201
Polička

Hledám kamaráda z Prahy (nad 18 let), který by me pomohl v programování ve strojaku. Jedna se o ružna intra a dema ve stylu DEMO DEPO.

Jiří Matejka, Bochovska 541, 158
00 Praha 5-Jinonice



Digital Sound Studio 8



Jistě vám všem dobře známá americká firma Great Valley Products Inc. uvádí na trh svůj nový produkt - Digital Sound Studio 8.

Jedná se o kvalitní stereo sound sampler a rychlý uživatelsky velice přívětivý zvukový editor, který obsahuje i čtyřstopý sekvencer typu X-tracker.

Hardware je černá krabička, připojitelná na paralelní port Amigy. Sampler má dva regulátory úrovně vstupního signálu, kontrolované LED

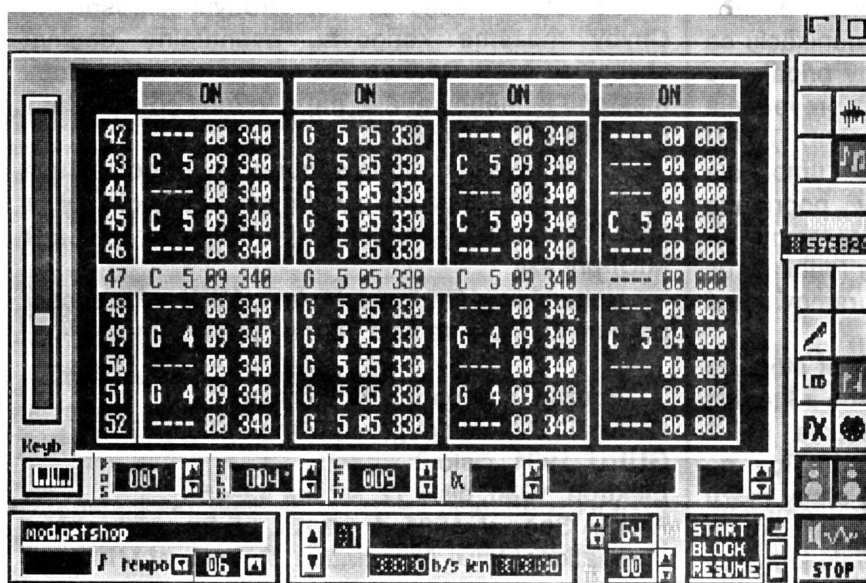
reálném čase. Dvě ikony pod jsou určeny pro monitoring a editaci samplovaných zvuků. Tři osminové noty jsou ikonou pro vstup do módu sekvenceru.

Editace samplovaných zvuků je špičková, neboť práce se samplý je v reálném čase. Do programu je vestavěno osm efektů kterými lze zvuky upravovat (např. echo, mix, filter, fade-in/out, atd.). Nasnímané zvuky je možné ukládat ve formátech IFF, SONIX nebo RAW v rozsahu 1, 3 nebo 5 oktáv.

Program samozřejmě pracuje jak v módu stereo, tak i mono. Na stereo módu je zajímavá maximální rychlost 51kHz. DSS zaujme také osciloskopem a spektrum analyzerem pracujícím v reálném čase.

Sekvencer je nedílnou součástí programu. Je kompatibilní s programy SoundTracker, NoiseTracker a SoundFX. Můžeme zde editovat hudební moduly těchto programů, i když ne tak komfortně, jako v samotných hudebních programech.

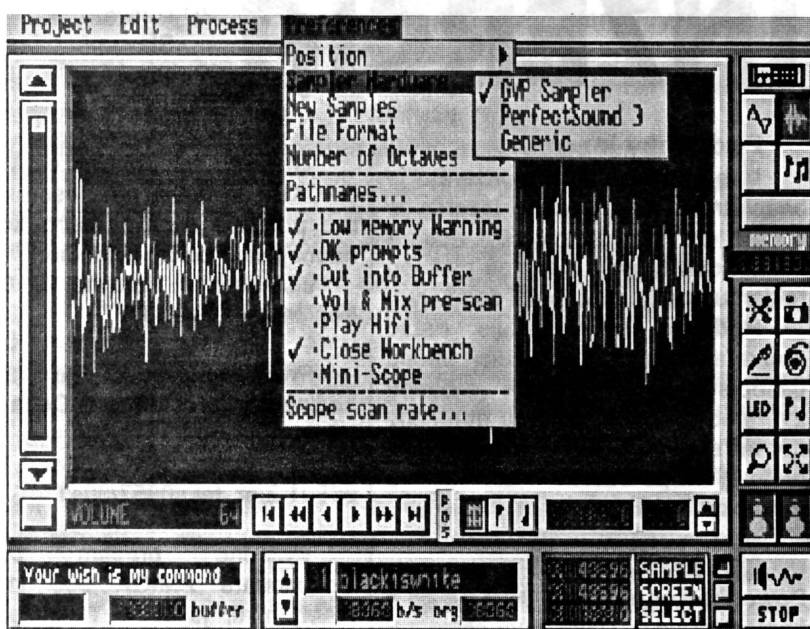
-lh-



diodami proti přebuzení pro každý kanál zvlášť. Vstup signálu je přes dva Cinch konektory.

Program je napsán v assembleru (proto ta rychlost) a je kompatibilní s Amiga DOS 2.0 a umožňuje multitasking, samozřejmě na počítačích s větší kapacitou paměti. Jak už je u GVP zvykem software i hardware je použitelný i na mikroprocesorech 68020 a 68030.

Velice pěkně je vyřešeno ovládání mezi sekvencerem a editorem. Po inicializaci se program přihlásí vestavěnou databankou zvuků, která má 31 položek. Je do ní možné uložit až 31 samplů současně (opět ta kapacita paměti). V pravo nahoře jsou umístěny čtyři ikony (u verze 1.0). Horní ikona slouží na nastavení parametrů dozvuku v





Tak to bys měl znát aneb nejen ke hrám jest Amiga předurčena

Vítám čtenáře u zrodu nového seriálu, který se bude snažit objasnit, pokus co nejsrozumitelnější cestou základní pojmy a akce týkající se naší společné přítelkyně. Tedy příležitost pro všechny mocné lamače joysticků, ale samozřejmě nejen pro ně. Byl bych rád, kdyby si každý z vás uvědomil po přečtení některé z částí tohoto seriálu, jaké mocné zbraně se vám v podobě Amigy dostalo do rukou. Ano, nejen ke hrám byla Amiga stvořena, třebaže tato slova zazní nelaskavě uším kdejakého playera.

Cím se tedy budeme v našem seriálu zabírat? Ze začátku se zaměříme na objasnění nejzákladnějších pojmů, s kterými se často pracuje. Chci, aby vám při takových pojmech jako CLI, Devices nebo "logický název" nevstávali hrůzovlas na hlavě. U cizojazyčných názvů si budeme rovněž uvádět jednak jejich správnou výslovnost (tedy něco ve formě maleho kurzu angličtiny) a jednak jejich překlad do češtiny. To všechno proto, aby se i našinec dokázal domluvit a to nejen na území Čech, Moravy a Slovenska.

Dalšími tématy budou například: systémová disketa, kde se podíváme na "správný" obsah diskety, abychom byli schopni sestavit potřebnou disketu se vším všudy. Podíváme se nato, jak naučit přítelkyně českému či slovenskému jazyku, jak správně obsluhovat tiskárnu apod. Tady bych chtěl podotknout, že tato rubrika není moje, ale naše. Tedy výše uvedenou náplň berte jen orientačně, protože v případě zájmu o nějaký problém z vaší strany vše přizpůsobíme. Směřujte tedy vaše směle dopisy na adresu redakce.

Toliko úvodem.

Dnes se podíváme na zavedení systémové diskety a mrkneme se i na nějaký ten pojem. Beru na vědomí, že je mezi vámi dosti věcí znalých, kterým tyto pojmy a vůbec vše dále uvedeno bude důvěrně známo. Myslím si ale, a mám to bohužel experimentálně potvrzeno, že nás věcí neznalých je stále ještě dosti. Je na čase udělati konec metodám pokusů a omylů. A je taktéž třeba zatrhnouti prodávání směšných informací za smutné peníze. Dosti však řečí a dejme se do díla.

Disketa to je první předmět, se kterým přišel každý z nás do styku hned po rozbalení počítače. Víme taky, že naše přítelkyně, i když jinak velmi chytrá, neumí bez diskety "naběhnout". Ale teď trochu blíže - z čeho je taková disketa tvořena?

Diskety pracují na podobném principu jako obyčejný magnetofonový pásek, tedy nic nového pod sluncem. Aby s ní však uměl počítač spolupracovat, je ji třeba naformátovat. Tedy převést ji do formátu, který počítač zná.

Formátování obnáší vymazání diskety spolu s vyznačením pomyslných kružnic na disketě. Těchto imaginárních kružnic je u standardní Amigy 160 a protože má disketa 2 strany, tedy 80 na každé straně. V našich odborných kruzích budeme nazývat tyto kružnice **STOPY** (anglicky **TRACKS** [čti treks]).

Určitě si každý z vás také všiml, že ne všechny diskety nám odstartují naši Amigu, třebaže jsou zformátovány. Za toto nese odpovědnost tzv. instalování diskety. Instalování způsobí, že se na část nulté stopy - na tzv. **BOOTBLOCK** [bu:tblok] запиše kratičkový program, který umožní zavedení diskety. Tento bootblock (nezapomeň na výslovnost) je právě místem, kde se často ukrývají různé viry.

Shrňme tedy naše poznatky: abychom mohli použít disketu v naší Amize, musíme ji nejprve zformátovat, pokud z ní také chceme "probudit" Amigu, pak ji také instalujeme (k tomuto slouží příkazy: **FORMAT** a **INSTALL**).

Například: disketa Workbench je jak zformátovaná, tak i instalovaná, naopak disketa Extras je jen zformátovaná.

Ještě technická poznámka: po zformátování je na 3.5" disketě (3.5" znamená tři a půl palcová) volných necelých 880kBytů.

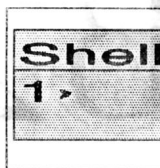
Co se však děje po "nabütování" (zavedení) diskety? Amiga se nejprve pokusí nahrát soubor "**SYSTEM-CONFIGURATION**", kde jsou uloženy barvy pro obrazovku a další informace, o kterých podrobněji pojednáme v některém z dalších pokračování (tento soubor se vytváří programem "Preferences"). A málem bych zapoměl na výslovnost, tedy "system-configuration" se čte [sistim konfigjurejšn], kde "ti" se nezměkčuje - resp. čte se jako "ty". A překládá se jako systémová konfigurace, neboli nastavení systému.

Jako druhý se nahrává soubor "**STARTUP-SEQUENCE**" [startap si:kwens] a možno jej přeložit jako startovací sekvence. Tento soubor patří mezi tzv. povelové soubory, o kterých si však povíme až příště.

Na další spolupráci se těší a hodně trpělivosti do další práce přeje

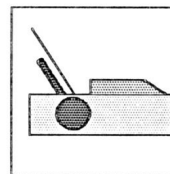
Jofa

PS: Prosím všechny angličtináře za prominutí mé zjednodušené výslovnosti, ale zde opravdu není místo na větší rozbor angličtiny. Thanx!





DOWNLOAD



Po dvou dílech našeho seriálu jste si již mohli udělat obrázek o tom, co by taková tiskárna měla umět. Dnes se zaměříme na další vlastnost tiskárny a tím je tzv. "DOWNLOAD" (dále DL).

Co to DL je? Pro neznalé asi tolik - je to možnost definovat souborem vyslaným do tiskárny vlastní sadu znaků. Tedy i naši abecedu. Jak na to? To se dozvíte právě teď.

Nejdříve musíme otevřít manuál, který jsme dostali při koupi k tiskárně a polistujeme na prvních pár listech, abychom zjistili, má-li naše tiskárna možnost přijmout DL. Tuto informaci najdeme buďto v obsahu, nejpravděpodobněji jako "Download Characters". Dále najdeme informace o popisu přepínačů DIP. Zde se podíváme do tabulky na položku "Usage of RAM - Buffer/Download". Tento přepínač musíme přepnout na Download. RAM paměť tiskárny nebude použita jako buffer, ale paměť předlohy námi definovaných znaků. Nesmíme také zapomenout přepnout tiskárnu do módu Standart, která emuluje řídicí kódy tiskáren Epson. V módu IBM nebude DL fungovat.

V našich podmínkách je velice rozšířena tiskárna LC-10 resp. LC-20, proto se zaměříme na popis vytvoření DLU pro tuto tiskárnu. Vytvořený DL by měl pracovat na všech devítijehličkových tiskárnách emulujících mód Epson.

Vytvoření a možnost tisku našich znaků vyžaduje dvě podmínky: poprvé - vytvoření předdefinovaných znaků a sestavení do souboru, který vyšleme do tiskárny. Podruhé - samotná

inicializace tiskárny a vyslání DL.

Tedy - naše LC-20 je devítijehličková tiskárna. Jeden znak je tvořen maticí devíti bodů na výšku a šesti body na šířku. To není ovšem vše - horizontálně mohou být jehly tiskárny umístěny ve středu čtverce nebo mezi čtverci na lince. Tímto je zvýšen počet bodů v rastru na 11. Vertikálně jehly tisknou pouze do středu čtverce (viz obrázek). Další podmínky definice jsou minimální šířka znaku 5 bodů a maximální 8 bodů vertikálně. A ještě musíme znát pozici námi definovaného (definovaných) znaku v ASCII tabulce.

Tedy začneme pohledem do ASCII tabulky znaků, abychom zjistili pozice znaků, které budeme definovat. Abychom dodrželi kompatibilitu, doporučuji držet se nějakého standartu - třeba norma KOI-8 (PageStream-cs, GURU). Porovnáním postavení znaků v KOI-8 a ASCII tabulce zjistíme, že budeme definovat znaky od C1 po FA (všechny hodnoty jsou hexadecimálně) - tedy od "á" až po "Z". Hodnotě C1 je standardně přiřazen znak "Á", ovšem my jej musíme předefinovat na á. Jak?

Nejdříve si nakreslíme na papír mřížku o rozměrech 6x8 jako na obrázku a stejně ji také označíme. Pak nakreslíme "á" do této mřížky. Nyní sečteme po sloupcích hodnoty m1 - m11, které jsme obsadili naším "á". Dostaneme hodnoty 00, 04, 0A, 20, 0A, 60, 8A, 20, 1C, 02, 00 (Hex). Každý definovaný znak má ještě tzv. "attribute" - m0, kterým definujeme pozici znaku. Je složený ze tří složek: Pozice, mezera zleva a poslední sloupec. Pozice určuje umístění znaku vzhledem k řádce - máme možnost definice znaku na řádek nebo pod (např. ý). Nabývá hodnot 1 nebo 0. Mezera zleva a poslední sloupec udávají šířku znaku, a tak dovolují precizně definovat umístění znaku v mřížce, proto např. úzké "i" bude vypadat atraktivněji vedle širších "m, o". Hodnoty pro mezeru zleva jsou akceptovány od 0 po 7, pro poslední sloupec je to od 4 do 11.

Vzorec výpočtu:

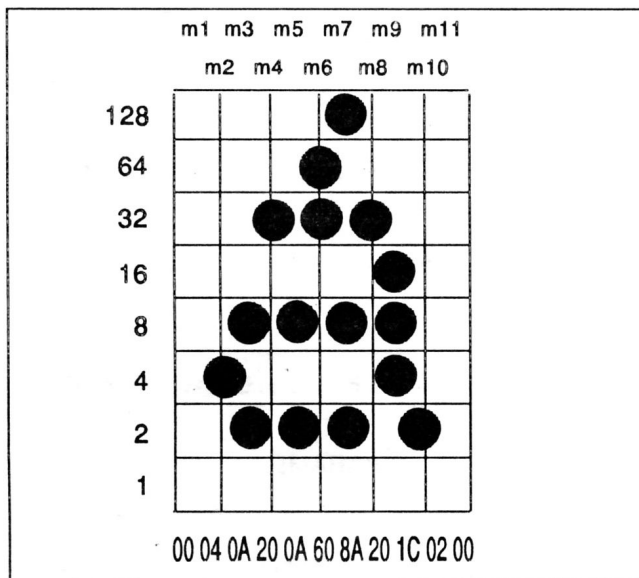
$m0 = (\text{Pozice} \times 128) + (\text{vlevo} \times 16) + \text{posl.sloupec}$

Podle vzorce spočítáme hodnotu pro "á": $m0 = 9B$. Přiřazením hodnoty 9B před hodnoty m1 až m11 dostáváme definici znaku "á".

Takto vytvoříme za pomoci mřížky celou naši znakovou sadu malých i velkých písmen s diakritikou.

Jistě jste si všimli v příloze příručky přehledu příkazů, kterými lze tiskárnu ovládat příkazem "Esc" a následnou sekvencí znaků (viz předchozí díl).

Tedy první příkaz, který pošleme na tiskárnu je



reset, abychom zamezili případným nesmyslům, které by asi znechutili naši usilovnou činnost. Tedy pohledem do manuálu zjistíme sekvenci resetu tiskárny: <ESC> "@" - 1B 40 (Hex).

Další parametr je zkopírování sady znaků z ROM do RAM paměti tiskárny: <ESC> "."<0> n <0> - 1B 3A 00 00 00 (Hex).

Nyní samotná definice "DL-u" <ESC> "&" <0> n1 n2 m0 m1 m11, kde n1 je ASCII kód prvního znak, který budeme předdefinovávat, n2 je poslední předdefinovaný znak (v našem případě tedy od C1 po FA), m0 - m11 jsou atributy a datové byty nového znaku. Tedy: 1B 26 00 C1 FA (Hex).

Na konec celého souboru pak musíme dát sekvenci výběru našeho "DL-u": <ESC> "%" "1" - 1B 25 31 (Hex).

Začneme tedy sestavovat DL: 1B 40 1B 3A 00 00 00 1B 26 00 C1 FA 9B 00 04 0A 20 0A 60 8A 20 1C 02 00

Nyní stojíme před problémem sestavení těchto kódů do souboru. Budeme na to potřebovat nějaký hexaeditor (FileMaster). Pro ty, kteří zvládnou jednoduché programování např. v AMIGA Basicu, doporučujeme spíše tento postup (viz listing).

Soubor DL-u si uložíme na disketu, např. pod názvem DL. Inicializace a tisk s čs. diakritikou probíhá následovně:

Nejdříve si nahrajeme jednoduchý textový editor, ve kterém používáme fonty shodující se s normou KOI-8 (nemáte-li, použijte GuruFont,9). Pro test našeho DL si napište soubor, který obsahuje celou naši znakovou sadu s diakritikou a uložte ho na disk jako "TEXTTEST". Tiskárna je připojena na paralelní port Amigy, proto musíme DL směřovat na tento port. DL tedy vyšleme do tiskárny ze Shell nebo CLI příkazem "TYPE"

TYPE DL TO PAR:

V případě, že je tiskárna zapnutá, DIP přepínač je přepnut na DL a papír máme zasunut - tedy vše je OK, papír by se měl po odeslání RETURN-em trochu posunout. To je znamení, že byl DL tiskárnou přijat a byl aktivován. Nyní do editoru nahrajeme soubor "TEXTTEST" a zkusíme jej vytisknout. Doporučuji směřovat soubor na PAR:, neboť nastavený driver tiskárny v PRT: může některé znaky změnit nebo je vůbec nepustí (blokuje). V případě neúspěchu vyšlete soubor z CLI opět příkazem TYPE.

TYPE TEXTTEST TO PAR:

Na tiskárně by se měly vytisknout znaky naší milé abecedy.

Kompletní DL pro tiskárny LC najdete v GURU 3/92

-lh-



Tabulka znaků s diakritikou v normě KOI_8

Dec	Hex	Znak			
			220	DC	
			221	DD	
193	C1	á	222	DE	
194	C2		223	DF	
195	C3	č	224	E0	
196	C4	ď	225	E1	Á
197	C5	ě	226	E2	
198	C6	ř	227	E3	Č
199	C7		228	E4	Ď
200	C8		229	E5	Ě
201	C9	í	230	E6	Ř
202	CA	ů	231	E7	
203	CB	í	232	E8	
204	CC	l'	233	E9	Í
205	CD		234	EA	Ů
206	CE	ň	235	EB	Ľ
207	CF	ó	236	EC	Ł
208	D0	ô	237	ED	
209	D1	ä	238	EE	Ń
210	D2	ř	239	EF	Ō
211	D3	š	240	F0	Ö
212	D4	ť	241	F1	Ä
213	D5	ú	242	F2	Ř
214	D6	é	243	F3	Š
215	D7		244	F4	Ť
216	D8	ý	245	F5	Ů
217	D9	ž	246	F6	É
218	DA		247	F7	
219	DB		248	F8	Ý
			249	F9	Ž

REM UVEDENE PARAMETRY PLATI PRO LC-20 (10)
PRE KOI-8

POCETPOLOZEK=? :REM SPRAVNY POCET
DOPLNIT

OPEN "R",#1,"RAM:DWL",1
FIELD #1,1 AS B\$

FOR I%=1 TO POCETPOLOZEK
READ A\$
A=VAL("&H"+A\$)
LSET B\$=RIGHT\$(MKI\$(A),1)
PUT #1,I%
NEXT I%

CLOSE #1
END

REM RESET TISKARNY
DATA 1B,40

REM ZKOPIROVANI CHARSETU Z ROM DO RAM
DATA 1B,3A,00,00,00

REM DEFINOVANIE PARAMETROV
DOWNLOADU
DATA 1B,26,00,C1,FA

REM VLASTNA DATA PISMEN



BOOT X

metla na vírusy

Teraz sa pozrieme do menu Bootblocks.

Load/Save BootX môže nahráť BB do pamäti z ľubovoľného platného Amiga-zariadenia. Položka 'Save BB' môže byť použitá na zálohovanie bootblokov hier, demo-programov a iných programov na samostatný data disk. V prípade, že vírus zničí BB vašej diskety, môžete ho načítať do pamäte a nahráť ho znova na originálny disk.

Load BB

umožní nahráť iba súbor, ktorý obsahuje skutočný bootblok. Kontroluje, či je starý alebo je to Fast File System bootblok, a potom prepočíta checksum (kontrolný súčet) bootbloku. Ak je kontrolný súčet bootbloku zlý, BootX sa vás opýta, či ho chcete opraviť. Ak zvolíte Yes, BootX opraví checksum nahratej kópie bootbloku.

Learn BB (nauč sa BB)

Kliknutím tohoto gadgetu máte možnosť zadať meno BB v pamäti, ak ho ešte BootX nepozná. Stlačte len klávesu Return a program potom doplní BB do aktuálneho Brainfile (ak je prítomný). Inak vytvorí v pamäti nový Brainfile.

Voľby menu Files

Check files (kontroluj súbory)

Voľba tejto položky vyvolá requester umožňujúci zvoliť si vaše vlastné nastavenie na kontrolu proti linkovým a súborovým vírusom.

- Skip directories (preskakuj adresáre)

Ak je zvolená táto voľba, BootX preskočí všetky podadresáre, ktoré nájde. Pomocou tejto voľby môžete rýchlo prezrieť iba súbory v aktuálnom adresári disku alebo harddisku.

- Check All Files (skontroluj všetky súbory)

BootX normálne skontroluje iba spustiteľné súbory (v technickom jazyku: súbory, ktoré majú platnú Amiga DOS štruktúru hunkov), keďže len tieto súbory sa môžu vykonať. Ak je zvolená táto voľba, BootX skontroluje každý súbor, ktorý nájde, dôsledkom čoho je celkom príjemné predídenie doby kontroly.

- Enter directories interactively (vstupuj interaktívne)

Ak je zvolená táto voľba, BootX sa zakaždým, keď nájde adresár opýta, či má doňho vstúpiť. Toto je pohodlné pri snímaní harddiskov s veľkým množstvom adresárov. Vypíše sa requester, ktorý ponúka možnosť vstúpiť do adresára (zvoľ gadget Yes), alebo adresár preskočiť (zvoľ gadget No).

- Auto-remove read-protection (automatické odstraňovanie ochrany)

AmigaDOS 2.0 rozoznáva a podporuje ochranné bity súboru. To znamená, že ak je zapnutá ochrana proti načítaniu, žiadny program tento súbor nenačíta. Keď BootX objaví takýto súbor opýta sa, či z neho chcete vymazať ochranu. Ak zvolíte Yes, BootX ponechá všetky ostatné ochranné bity okrem bitu ochrany proti načítaniu. Ochrana proti načítaniu bude vypnutá a súbor prekontrolovaný.

Ak by ste chceli obnoviť ochranu proti načítaniu, použijete príkaz AmigaDOSu Protect:

1> protect <názov súboru> -r

Ak zvolíte túto voľbu, položka menu "Auto skip" bude vypnutá (viď ďalej).

- Auto-skip read-protected files (automatické preskakovanie chránených súborov)

Niekedy sa môže stať, že chcete ochrániť určité súbory proti načítaniu prepísaním ochranného bitu. Keď je zvolená táto voľba, BootX bude ignorovať všetky súbory s ochranou proti načítaniu. Ak zvolíte tento gadget, položka auto-remove bude vypnutá (viď vyššie).

- Handle viruses automatically (vírusy odstraňuj automaticky)

Ak BootX zistí vírus, opýta sa, či má použiť niektorú z troch ponúkaných metód. Túto voľbu zvolte, ak chcete, aby BootX vírusy odstraňoval ihneď po ich zistení bez opýtania.

Poznámka: posledné dve voľby sú vhodné najmä pri prezeraní harddiskov s veľkým množstvom súborov.

- Virus Mark options (voľby označovania vírusov)

Môžete si vybrať niektorý z troch spôsobov, ako má BootX zaobchádzať s vírusmi. K súboru môžete

doplniť poznámku, vymazať ho alebo premenovať. Ak BootX premenúva súbor, použije názov, ktorý je zložený z názvu vírusu a špeciálneho počítačľa, ktoré zabezpečuje špecifickosť všetkých názvov vírusov. **Filenote** (poznámka) alebo **Comment** (komentár) doplnia k súboru určité špecifiká - názov vírusu a verziu BootXu, ktorá ho našla. Tento komentár je viditeľný pri použití príkazu "List" z CLI alebo Shell.

- Report features (rysy reportu)

Tieto voľby umožnia vytvoriť prehľad vášho disku alebo harddisku. Po skončení kontroly sa BootX opýta, čo má s prehľadom spraviť. Môžete sa doň buď pozrieť, alebo ho uschovať na disketu. Normálne BootX zobrazuje len infikované súbory alebo súbory, v ktorých sú chyby.

Keď zapnete voľbu "**Report file types**" (zobrazujú typy súborov), program do reportu pridá každý súbor, ktorý nájde, spolu s označením typu.

- Check files (kontroluj súbory)

Tento gadget vyvolá requester, v ktorom máte upresniť zariadenie, kde sa budú linkové a súborové vírusy hľadať. Môžete zvoliť akékoľvek AmigaDOS zariadenie, ktoré podporuje súborový systém. Keď zvolíte OK, BootX začne kontrolovať súbory jeden po druhom tak, ako ich nájde, pričom však stále berie na vedomie zvolené preferencie.

- Show report again (zobraz report ešte raz)

Táto položka menu umožní znova si prezrieť naposledy vytvorený report.

- Save report (uschovej report)

Táto položka menu umožní naposledy vytvorený report uschovať na disk.

- Check Disk (skontroluj Disk)

Ak zvolíte túto položku menu, BootX vás vyzve, aby ste do aktuálnej mechaniky vložili disk. Následne vykoná všetky kontroly, čo znamená, že najprv bude skontrolovaný bootblok a potom aj disk-validator. Nato prebehne snímanie celého disku na infikované sektory. Na záver budú skontrolované všetky súbory. Prirodzene, BootX sa opýta, či si budete želať preskočiť niektorý z

vyššie menovaných troch spôsobov kontroly.

- Repair disk (oprav disk)

Táto položka menu umožní opraviť poškodenia spôsobené vírusom "Saddam", ktorý sídli v disk-validatore. Tento vírus poškodí bitmap disku a zakóduje databloky na diskete. Pokiaľ je vírus v pamäti, tieto bloky sú rozkódované vždy, ak sú čítané. Teda disketa sa správa normálne, je čitateľná. Ale ak disk používate na neinfikovanej Amige, tieto bloky sa stanú nepoužiteľné. Repair Disk zosníma disk a pokúsi sa napraviť poškodenia. Presvedčte sa, či v pamäti nie je žiaden vírus (predovšetkým Saddam vírus). BootX kvôli istote skontroluje pamäť a v prípade pozitívneho nálezu vírus vymaže. BootX potom zamkne aktuálnu mechaniku (zvoľ aktuálnu ktorúkoľvek mechaniku použitím 4 gadgetov mechaník popísaných vyššie) a vyzve vás vložiť poškodený disk. Počas snímania dostanete aktuálny report. Na diskete môžete kludne nechať "otvorené okienko" (nechať ho zaistený proti zapisu), disk bude iba zosnímaný bez akýchkoľvek úprav.

Ako BootX ničí linkové a súborové vírusy... ? Ak dáte programu BootX skontrolovať disk alebo adresár, BootX začne snímať a kontrolovať každý súbor, ktorý nájde. Keď nájde súbor s ochranou proti načítaniu, BootX sa opýta, či si prajete zrušiť túto ochranu. Ak zvolíte No, BootX tento súbor ignoruje a nekontroluje ho. V prípade, že zvolíte

BootX v4.13 (9.11.91)

Quit Read BB Write BB (DF0:)DF1: DF2: DF3: Lock drive
Install < > Anti-Virus 5.0 @ Peter Stuer

BootX v4.13 by P. Stuer - FREeware
Released on November 9, 1991

- o Recognizes over 385 bootblocks (159+ bootblock viruses and their mutants or family members)
- o Written in 100% machine code for lightning speed
- o Easy to use interface, Kickstart 2.0 compatible
- o Can be run from the Workbench or a CLI
- o Regularly updated (Look out for new bootblock libraries)
- o Recognizes 29 file or linkviruses
- o Thanks to Erik Løvendahl Sørensen for his Virus Bank and to Nico François, Bart Leeten and Didier De Cock for Beta testing.
- o Send new boot-, file- and linkviruses to:
Peter Stuer, Kauwlei 21, B-2550 Kontich, Belgium - Europe
- o Spread the program, not the word !

gadget Yes, BootX sa pokúsi zrušiť ochranu proti načítaniu. Súbor je potom kontrolovaný, aby sa ukázalo, či obsahuje vírus. Či už infikovaný súbor označíte alebo nie, BootX nastaví ochranné bity na "---A---D", čo znamená, že súbor môže byť iba zmazaný. Bit Archive zaručuje, že súbor nebude obsiahnutý v záložnej kópii adresára.

Menu Miscellaneous

Check Memory (skontroluj pamäť)

Kontroluje pamäť na známe vírusy, pričom ignoruje známe antivírusové programy. Ak BootX nájde vírus, vypíše oznámenie s identifikáciou vírusu a pokúsi sa ho odstrániť. BootX tiež vymaže z pamäti všetky známe linkové vírusy. Po kontrole pamäti BootX zobrazí niektoré dôležité vektory z ExecBase. Tieto vektory sú najčastejšie menené vírusmi. Normálne vektory budú zobrazené čiernou farbou. Ak si BootX myslí, že niečo nie je v poriadku, pri vektore zobrazí odkaz s bielymi písmenami: "Pleasecheck...".

Uvedomte si, že to hneď neznamena, že v pamäti je vírus. Rezidentné programy ako napr. TurboPrint tieto vektory používajú tiež.

BootX bude aktualizovaný v čo najkratšom čase. V prípade pochybností nahrajte všetky data a preveďte studený štart, čo znamená Amigu na niekoľko sekúnd vypnúť.

Výpis obsahuje ešte mnoho ďalších technických informácií. Ak ste začiatok a BootX vypíše, že niektoré z vektorov boli zmenené a nevíete, ktorý program ich zmenil, poraďte sa u skúsenejšieho užívateľa (najlepšie u programátora).

Reset Vectors (resetuj vektory)

Tento gadget dáva možnosť zmazať prvých päť vektorov, ktoré BootX zobrazuje vo vektorovom listingu. Toto však nie je zaručený spôsob likvidácie vírusu!

Install resident LVD (nainštaluj rezidentný LVD)

LVD je program, ktorý kontroluje kľúčové body a vektory operačného systému Amigy. Po inštalácii nespotrebuje žiaden čas procesoru, ale len do tej chvíle, pokiaľ sa nepokúsite nahráť nejaký spustiteľný súbor (program). Predtým, než sa program spustí, LVD skontroluje, či v ňom nie je ukrytý nejaký známy linkový vírus. Ak je kód programu zmenený, program nebude spustený. Toto je preventívna ochrana pred aktiváciou vírusu. Ale ak už máte v pamäti vírus, každý program, ktorý nahráte, môže byť infikovaný. Táto voľba menu

nainštaluje rezidentnú verziu LVD s prístupom na akýkoľvek disk. Najprv sa presvedčte, či nie je v pamäti vírus (**Check Memory a Reset vectors**). Potom nainštalujte program LVD. LVD je resetu vzdorný, teda po resetovaní počítača zostane naďalej v pamäti. Teraz ak je v startup-sequence názov infikovaného programu, LVD ho zlikviduje skôr, než sa rozbehne.

POZNÁMKA: LVD sa nainštaluje na koniec KickTag-u. Takto, ak nejaký iný program používa tento vektor, nechá ho aktívny.

Voľby menu Settings

Detection diskchange (zisťovanie výmeny disku)

Táto položka menu pracuje ako vypínač. Vypína alebo zapína detekciu výmeny disku. Niekedy nie je vhodné, aby BootX načítal BB každého vloženého disku, obzvlášť keď chcete nahráť BB, prítomný v pamäti, na inú disketu. Tento gadget umožňuje užívateľom jednej mechaniky kopírovať bootbloky.

Show Help (zobraz pomoc)

Táto položka menu zapína alebo vypína voľbu **Help** (pomoc). Ak je zapnutá, môžete kliknúť na akýkoľvek gadget alebo položku menu a zobrazí sa stručný popis jej funkcie.

Pause after page (po stránke prestávka)

Táto voľba kontroluje, či vypisovanie textu bude zastavené po zaplnení stránky (resp. monitoru). Táto voľba je dočasne vypnutá, ak ste zvolili kontrolu súborov.

Check Memory at startup (pri startupe kontroluj pamäť)

Keď je zvolená táto voľba, BootX vykoná kompletnú kontrolu pamäte vždy po spustení alebo po "zobudení".

Check Disk-Validator (kontroluj Disk-Validator)

Keď vložíte disk do aktuálnej mechaniky, BootX sa najprv pokúsi identifikovať bootblok. Ak je táto voľba zapnutá, skontroluje aj disk-validator. Takto môžete spoľahlivo odhaliť disk-validatorové vírusy skôr, než sa spustia.

Poznámka: táto voľba je obzvlášť účinná, ak je aktuálna mechanika zamknutá.

Použitím anglickej dokumentácie spracoval
Ing. Miroslav Tomeček.

Formátovací kódy programu WRITE

Znáte textový editor WRITE z disku "Appetizer"? Cože? Že je hloupý, nezná hlavičky a patičky, nedá se v něm nic nastavit a navíc je německý! Chyba lávky přátelé! Pomocí formátovacích kódů můžete dostat i z tohoto jednoduchého, nikoli však hloupého, textového editoru to, co většina z vás potřebuje.

Write v sobě obsahuje všechny základní funkce co by měl mít pro psaní dopisů, dokumentů, manuálů atd.. Je jednoduchý na ovládání. Jako jediný má zvukovou odezvu při psaní, což je velice důležité pokud jste zvyklí psát všemi deseti prsty. A už vás slyším "Copak píšu všemi deseti prsty? Stačí dva a hotovo!". Vy možná ne, ale kdokoli, jenž využívá textový editor deňně k psaní textů, si bez této funkce nemůže sluchem kontrolovat, zda klávesu už zmáčkli či nikoli. Stále šmejdít znakem z předlohy na obrazovku a zpět, to utahá i odolného písáče, natož vás. Další výhodou je dosažitelnost funkcí z ikon na kraji obrazovky. Můžete tisknout zvláště liché a sudé strany, což znamená možnost oboustranného tisku. Tisk můžeme provést buď na tiskárnu nebo do souboru.

A tak bych mohl pokračovat dále, ale já bych vás chtěl seznámit spíše s tím, co nenajdete v horním menu ani v ikonách. Jsou to formátovací kódy pomocí nichž můžete formovat výstup textu do souboru nebo na tiskárnu.

Z čeho se skládá formátovací kód?

Na prvním místě stojí vždy znak "\$". Vyvoláme ho současným zmáčknutím kláves <SHIFT><ALT><S>. Pokud máte německou klávesnici, vyvoláte jej zmáčknutím kláves <SHIFT><3>. Těsně za tímto znakem stojí řídící kód, např. jc,lr,ll,tm,jf atd.. Co tyto kódy znamenají si vysvětlíme za chvíli. Za těmito dvěma znaky stojí číslo, které budeme obecně značit <n>. "n" představuje nastavovanou hodnotu funkce. Uvedu pár příkladů pro názornost "\$jc", "\$lb10", "\$tm0". První příklad vycentruje text, který se nachází za ním. Druhý nastaví levý okraj pro všechny strany na 10 znaků od okraje papíru. Třetí příklad nastaví horní okraj papíru na 0 řádků.

Jak používat formátovací kódy.

Aby WRITE poznal formátovací kód, musí stát vždy na prvním místě nového řádku. Pokud se vyskytuje kdekoli jinde, je ignorován. Je to proto, že pokud by jste psali nějaký právnícký text, nastal by totální zmatek. Takto pouze stačí dát si pozor, aby znak nebyl na prvním místě řádku, a je po problému.

Jedním řádkem se rozumí řádek fyzický, tedy který je ukončený znakem <RETURN>. Více kódů můžeme psát na jeden řádek, ale musíme je oddělit dvojtečkou Např. takto "\$jc:ll10:tm0".

Význam jednotlivých formátovacích kódů.

Centrovací kódy:

jc - Text následující za tímto znakem je vycentrován doprostřed řádku. Toto nastavení platí tak dlouho, dokud se nevyskytne "jl", "jr", "jf".

jl - Text se zarovnává od levého okraje strany. Ruší "jr", "jc", "jf".

jr - Text se zarovnává od pravého okraje strany. Ruší "jl", "jc", "jf".

jf - Text je rovnán na pravý okraj. To znamená, že pravý okraj je stejně rovný jako levý. Nic nepřechází, ani nechýbí.

Nastavení okrajů strany:

lm<n> - Nastaví levý okraj strany pro obě strany. Tedy jak pro pravou (lichou) tak levou (sudou).

lb<n> - Nastaví levý okraj strany pro obě strany. Tedy jak pro pravou (lichou) tak levou (sudou).

ll<n> - Nastaví levý okraj strany pro levou (sudou) stranu (2,4,6...).

lr<n> - Nastaví levý okraj strany pro pravou (lichou) stranu (1,3,5...).

rm<n> - Nastaví pravý okraj strany pro obě strany. Tedy jak pro pravou, tak pro levou stranu.

rb<n> - Nastaví pravý okraj strany pro obě strany. Tedy jak pro pravou, tak pro levou stranu.

rl<n> - Nastaví pravý okraj strany pro levou (sudou) stranu (2,4,6...).

rr<n> - Nastaví pravý okraj strany pro pravou (lichou) stranu (1,3,5...).

Dokončení příště

Pep

!! PREDPOSLEDNÉ KOLO !!

SÚŤAŽ SÚŤAŽ

Vítame Vás v štvrtom kole našej súťaže, kde môžete vyhrať mnoho zaujímavých cien.

Pre zopakovanie znova uvedieme podmienky súťaže:

- zúčastniť sa môže **každý** odberateľ časopisu

- správne odpovedať na tri otázky, ktoré nájdete v každom čísle časopisu

- vyplnený kupón so správnymi odpoveďami zašli v obálke, alebo nalep na korešpondenčný lístok a do stanoveného dátumu odošli na adresu redakcie.

Súťaž trvá **päť** mesiacov a v prípade, ak správne odpovieš na všetkých 15 otázok, budeš zaradený do zlosovania, ktoré sa uskutoční v **júli (července) 1992**.

Budú vylosovaní **taja** šťastlivci, ktorí obdržia hodnotné ceny:

1. HARDDISK A 590 20MB

2. Externá disketová jednotka

3. Sampler A-MAX

Tí, ktorí zašlú kupóny, ale odpovede nebudú správne, nech nezúfajú. Aj zpmedzi nich vylosujeme troch, ktorí dostanú po krabičke diskiet.

Súťažné otázky štvrtého kola:

1. Ktorá rovnica vykreslí Lissajousovu krivku znázornenú na strane č.7 vpravo dole ?

- a) $X=R*\sin(\pi*3)+SX$ $Y=R*\sin(\pi*3)+SY$
b) $X=R*\sin(\pi*3)+SX$ $Y=R*\sin(\pi*5)+SY$
c) $X=R*\sin(\pi*4)+SX$ $Y=R*\sin(\pi*5)+SY$

2. Aká je prenosová rýchlosť v MIDI systéme?

- a) 3096.25 bytes/sec.
b) 3125 bytes/sec.
c) 4000 bytes/sec.

3. Riadiaci príkaz ESC pre tlačiarne má kód:

- a) \$1A
b) \$1B
c) \$1C



Prosíme, kupón vyplňte čitateľne a správne odpovede nezabudnite vyznačiť. Termín zaslania kupónu: **25. jún 1992** (rozhoduje podanie na pošte).

Veľa šťastia vám želá redakcia

SÚŤAŽ

Súťažný kupón štvrtého kola

1.	A	B	C
2.	A	B	C
3.	A	B	C

Meno a priezvisko:

Ulica:

PSČ, mesto:

**Zašlite najneskôr do
25. júna 1992**

Súťažný kupón štvrtého kola

SÚŤAŽ

!! PREDPOSLEDNÉ KOLO !!

Vážený příteli počítače AMIGA !

Prostřednictvím tohoto informačního letáku si Tě dovoluujeme upozornit na publikace vydávané námi :

Floppy magazín GURU

Je určen výlučně uživatelům počítačů Commodore AMIGA 500 - 3000. Jedná se již o druhý ročník úspěšného disketového magazínu, distribuovaného na 3.5" disketách. Disketa jako médium přináší oproti klasickému papíru celou řadu výhod a již osvědčených možností: listingy, hudba, grafika, demo,... . Vše je řízeno komfortním obslužným programem, který dovoluje vytisknout články podle potřeb uživatele na tiskárně.

Cena jednoho exempláře je 40.- Kčs včetně diskety noname a poštovného. Při použití značkové diskety jako média je cena 50.- Kčs.

V roce 1992 vyjde nejméně 6 čísel GURU, tedy v odstupu dvou měsíců. Při objednávce celého ročníku GURU poskytujeme slevu z 240.- na 219.- Kčs (noname) a při značkové disketě ze 300.- na 279.- Kčs.

AMIGA star

Jedná se o tištěný měsíčník na jehož 32 stranách najdete popisy PD programů, zaměříme se na zatím opomíjené oblasti zájmů (grafika, hudba, Ray-Tracking, programování). Vědomě potlačujeme rubriku her.

Cena jednoho čísla je 19.- Kčs, přičemž poskytujeme slevy na půl a celoroční předplatné:

na 6 čísel činí 105.- z původních 114.- Kčs na 12 čísel činí 199.- z původních 228.- Kčs

GURU i AMIGA star,

jejich média - disketa a papír, mají svoje klady, ale i zápory. Proto se budeme snažit, aby se disketa a časopis ve svých kladech vzájemně umocňovali a zápory se úplně eliminovaly. To docílíme vzájemnou provázaností obou médií a vzájemnou podporou. Popisy programů, tipy a triky se lépe čtou na papíře, ale listingy programů bez chyb nepřepíšete, proto budou na disketě.

Pokud se rozhodnete předplatit oba dva časopisy, nabízíme Vám výraznější slevy:

půlroční předplatné (6 x AMIGA star + 3 x GURU)

GURU noname činí 205.- z původních 234.- Kčs, GURU značk. disk činí 235.- z původních 264.- Kčs

celoroční předplatné (12 x AMIGA star + 6 x GURU)

GURU noname činí 405.- z původních 468.- Kčs, GURU značk. disk činí 475.- z původních 528.- Kčs

OBJEDNÁVKOVÝ KUPON

Objednávám si předplatné:

- časopisu **AMIGA star**

- floppy magazínu **GURU**

1 2 3 4 5 6 7 8 9 10 11 12

--	--	--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--

Disketa: noname značková

Částku Kčs jsem uhradil
složenkou typu C na adresu:

Eurofarm s.r.o.
Distribúcia
Dulovce
946 56

Jméno

Adresa

.....
.....

Datum a podpis

Hodící se vyznačte !

Hodící se vyznačte !

Firma

MIK

ponúka

Sound sampler

MIX 2.0

pre všetky modely počítačov

AMIGA 500 – 3000

Sampler sa pripája na paralelný port Amigy, vstup je prispôsobený pre najčastejšie používané zdroje nf signálu (magnetofóny, tunery, CD, syntetizéry). Spolupracuje so všetkými samplovacími programami aj pri najvyšších vzorkovacích frekvenciách (**56kHz**). Vysoká kvalita sampléru je zabezpečená vďaka obvodu **ZN 427**.

Na výrobok je poskytovaná **6 mesačná záruka** a je zaistený **záručný aj pozáručný servis**.

Cena: 790.- Kčs

Objednávky na adrese:

MIK

Rybárska 30
947 01 Hurbanovo

