amiga

cena 8 kčs *měsíčník pro uživatele počítačů amiga* *2/90*

Náš časopis a Amiga kluby

Když jsem napsal do svého úvodníku v nultém čísle, že Vám rádi objasníme Vaše nejasnosti kolem počítačů Amiga, netušil jsem, že budeme zaplaveni dopisy, ve kterých nám píšete jak Vás ten či ten klub "podvedl". Protože těch dopisů bylo skutečně mnoho, chtěl bych věnovat tomuto tématu následující prostor.

Je smutné, že mezi uživateli počítačů Amiga se najdou lidé, kteří rádi využívají neznalostí a nezkušeností začínajících uživatelů proto, aby se mohli nějak obohatit. Zakládají kluby s vysokým členským příspěvkem, za který slibují přednášky, literaturu a pomoc uživatelům. Bohužel, za nějakou dobu uživatel zjistí, že např. nedostal členskou legitimaci nebo že mu funkcionář klubu napíše "nezlobte se na nás, ale nemohu pro Vás nic udělat". A teď, uživateli, ukaž, co umíš. Bohužel, v této situaci Vám nemůžeme pomoci ani my.

Proto jsme se rozhodli, že od příštího čísla začneme uveřejňovat adresy klubů, které nedostáli svým závazkům. Na druhou stranu, budeme uveřejňovat též adresy klubů, které mají spokojenou členskou základnu. A teď to nejdůležitější: Jste-li členem nějakého Amiga klubu, napište nám své zkušenosti, abychom s nimi mohli seznámit i ostatní čtenáře.

S pozdravem

Khaled Husseini
šéfredaktor

O systémových knihovnách se v základních dvou příručkách na Amigu nedozvíme téměř nic. Co jsou to knihovny? O každé knihovně můžeme říci, že je to soubor funkcí, vztahujících se k jedné oblasti počítače (obraz, práce s disketou atd.). Knihovny se nacházejí jak v paměti ROM, tak i na systémových disketách v adresáři LIBS. Název knihovny je zakončen příponou .library.

Aby mohly být knihovny využity v AmigaBasicu, musí jeho interpretr "vědět", kde jsou jednotlivé funkce v knihovnách uloženy a jaká je jejich přesná syntaxe. Tyto informace najde v tzv. BMAP souborech. Tutíž ke každé knihovně patří jeden soubor se jmeném zakončeným .bmap. (Např. intuition.library - intuition.bmap.) Tyto soubory naleznete např. na disketě EXTRAS v adresáři BasicDemos. Tak jednoduché to však není! BMAP souborů je zde pouze několik pro nejpoužívanější knihovny. Ostatní si musíte nechat vytvořit AmigaBasic-programem ConvertFD, který je v adresáři Basicdemos. Ten je vytvoří z tzv. Fd souborů z adresáře FD1.2, resp. FD1.3 na disketě Extras. Po spuštění zadejte adresář a jméno zdrojového Fd souboru zakončené _lib.fd a to samé u cílového BMAP souboru. (Např. FD1.3/exec_lib.fd, BasicDemos/exec.bmap.) Program vám Bmap soubor vytvoří a uloží do zadaného místa. (Pro bližší informace viz AMIGA M&T 3/'90 čl. Wo sind die BMAPs.)

Ve svém programu musíme použití knihoven ozná příkazem LIBRARY, a to nejlépe na začátku. AmigaBasic si potom automaticky nahraje příslušný Bmap soubor z aktuálního adresáře.

Následující část programu ukazuje zavedení knihoven dos.library, intuition.library a graphics.library, jsou-li příslušné Bmap soubory na disketě Extras v adresáři BasicDemos.

```
10 CHDIR "Extras1.2D/BasicDemos"   ;nastavení aktual. adr.
20 LIBRARY "dos.library"           ;zavedení knihoven
30 LIBRARY "intuition.library"
40 LIBRARY "graphics.library"
```

Pro volání jednotlivých funkcí z knihoven se používá následující syntaxe: CALL jméno_funkce (parametry). Např. chceme-li použít funkce Delay() (z knihovny dos.library), syntaxe bude takováto:

```
50 CALL Delay(250)
```

Funkce Delay() pozastaví program po dobu zadanou v padesátinách sekundy. Na řádku 50 bude tedy program pět sekund čekat. Tento způsob volání nemůžeme použít v případě, kdy funkce vrací nějakou hodnotu. Pak musíme funkci nejdříve deklarovat. Př. pro funkci AllocMem() z knihovny exec.library:

```
45 DECLARE FUNCTION AllocMem() LIBRARY
60 adr=AllocMem(256,1004)
```

Funkce AllocMem() nám přidělí oblast paměti velikosti 256 bytů a jeho počáteční adresu uloží do proměnné adr. Druhý parametr specifikuje požadavky na paměť, to však teď není důležité.

intuition.library a graphics.library

O obou těchto knihovnách je možno říci, že provádějí operace s obrazem a s grafikou. To je pro programátory v AmigaBasicu důležité, protože AmigaBasic sám složitější operace s obrazem neumožňuje. Pro použití těchto knihoven je však potřeba znát strukturu a způsob zobrazování na počítačích Amiga.

Základními prvky jsou okna (window) a obrazovky (screen). O každém okně i o každé obrazovce si musí systém

IMPRESSUM

(c) 1990 "amiga – měsíčník pro uživatele počítačů amiga." Nevyžádané příspěvky se nevracejí.
Ročník I. Cena 8 Kčs. Reg. č. MK ČR 5281. MIČ 46067.
Vydavatel Jiří Prózr
Šéfredaktor: Khaled Husseini Odpovědný redaktor: Tomáš Stibor
Fotografie: Martin Kout Layout: redakce časopisu amiga
Tisk: SOF, Praha Sazba AZ servis Praha, tel.: 74 15 76
Redakce: Italské 29, 120 00 Praha 2 Inzerce, informace a předplatné: Italská 29, 120 00 Praha 2

uložit do paměti informace, aby s nimi mohl pracovat. Zde je nutné seznámit se s několika termíny.

Window structure, Screen structure - oblast paměti, v němž jsou uloženy o okně, resp. obrazovce, základní informace. Pokud otevíráme okno nebo obrazovku systémovými funkcemi OpenWindow(), resp. OpenScreen(), je nám vrácena adresa (tzv. ukazatel) struktury. Použijeme-li však příkazů AmigaBasicu na otevírání oken, resp. obrazovek, nevrátí nám tyto příkazy žádnou hodnotu. Pro zjištění ukazatele na strukturu okna je možno použít funkce AmigaBasicu Window(7).

RastPort - obsahuje data pro grafické a textové výstupy.

```
10 CHDIR "DF0:BasicDemos"
20 LIBRARY "intuiton.library"
30 GOSUB adresy
40 WBScreen&=sScreen&
50 GOSUB start
60 CALL WBenchToFront
70 FOR n=1 TO 150
80 CALL MoveScreen(WBScreen&,0,1)
90 NEXT
100 BEEP:BEEP:BEEP
110 GOSUB adresy
```

AmigaBasic a knihovny

Filip Železný

RastPort výstupního okna AmigaBasicu zjistíme pomocí funkce WINDOW(8).

ViewPort - informace o celkovém zobrazení (šířka, výška atd.)

I obraz na obrazovce musí být samozřejmě někde v paměti zaznamenaný. Princip spočívá v tom, že každý bod na obrazovce odpovídá jednomu bitu v paměti - je-li bit nastaven (má hodnotu jedna), pak je bod "rozsvícen" a naopak. Části paměti, odpovídající obrazu na obrazovce, se říká BitPlane. Tak by šel realizovat pouze jednobarevný obraz. Více barev získáme naskládáním více BitPlanes pro jeden obraz a barvy vznikají jejich kombinacemi. Tedy např.: Když je bod s jistými souřadnicemi zaznamenán v BitPlane č.1 jako nula (0), v BitPlane č.2 jako jednička (1) a v BitPlane č.3 také jako jednička (1), bude zobrazen barvou, odpovídající této kombinaci.

V předávání parametrů funkcím je ještě jedna věc důležitá. Pokud chceme funkci předat textový řetězec, nelze to udělat takto:

```
80 CALL Text(WINDOW(8),"ZDAR!",5)
```

Funkcím se nepředává přímo textový řetězec, ale ukazatel na něj. Ten zjistíme pomocí funkce SADD(řetězec$). Tedy:

```
80 a$="HAHA!"
90 adr=SADD(a$)
```

Do proměnné adr se umístí adresa řetězce a$ v paměti. Správně zapsaná funkce bude tedy vypadat takto:

```
80 CALL Text(WINDOW(8),SADD("ZDAR!"),5)
```

Funkce Text() z graphics.library napíše do okna s udaným RastPort (WINDOW(8)) řetězec o délce pěti znaků.

Některé funkce intuition.library

ScreenToBack(ScrStruct)/ScreenToFront(ScrStruct)
– Tyto funkce přemístí do pozadí, resp. do popředí, obrazovku s udaným ukazatelem na svou strukturu.

OpenWorkBench()/CloseWorkBench()
– První funkce otevře obrazovku WB. Druhá ji uzavře. Úspěšnost těchto funkcí je podmíněna tím, zda jsou či nejsou v užívání služby WB.

MoveScreen(ScrStruct,x,y)
– Pomocí této funkce je možno pohybovat s obrazovkou, udáte-li relativní souřadnice.

SetWindowTitles(WinStruct,WinTitle,ScrTitle)
– Nastaví název okna a obrazovky. Textové řetězce, obsahující názvy, musí být ukončeny znakem s ASCII-kódem 0. Názvy tedy nastavíme takto:

```
90 w$="JMÉNO OKNA"+CHR$(0):s$="JMÉNO
OBRAZOVKY"+CHR$(0)
100 CALL
SetWindowTitles(WINDOW(7),SADD(w$),SADD(s$))
```

SizeWindow(WinStruct,x,y)
– Nastavuje velikost okna. Souřadnice jsou relativní k stávající velikosti okna.

WBenchToBack()/WBenchToFront()
– Přemístí WB do pozadí, resp. do popředí.

WindowToBack(WinStruct)/WindowToFront(WinStruct)
– Přemístí okno do pozadí, resp. do popředí. Parametrem obou funkcí je ukazatel na strukturu okna.

Využití některých funkcí si ukážeme na programu.

```
120
w$="WINDOW"+CHR$(0):s$="SCREEN"+CHR$(0)
130 CALL
SetWindowTitles(WINDOW(7),SADD(w$),SADD(s$))
140 FOR n=1 TO 20
150 CALL MoveScreen(sScreen&,0,2)
160 CALL MoveScreen(sScreen&,0,-2)
170 NEXT
180 For n=1 TO 10
190  PRINT "**********"
200 NEXT
210 FOR n=1 TO 10
220  CALL MoveWindow(WINDOW(7),50,0)
230 NEXT
240 poc=640*150/8250:byte=1789788160
250 FOR n=BPlane1& TO BPlane1&+poc STEP 8
260  POKEL n,byte
270 NEXT
280 FOR n=1 TO 150 STEP 2
290 CALL MoveScreen(WBScreen&,0,-2)
300 NEXT
310 SCREEN CLOSE 1
320 END
adresy:
      sWindow&          = WINDOW(7)
      sScreen&          = PEEKL(sWindow& + 46)
      sRastPort&        = sScreen& + 84
      SBitMap&          = PEEKL(sRastPort& + 4)
      BPlane1&          = PEEKL(sBitMap& + 12)
RETURN
start:
      SCREEN 1,640,200,3,2
      WINDOW 1,,(50,50)-(100,100),0,1
      PALETTE 1,0,0,0
      PALETTE 0,.6,.5,.2
      PALETTE 7,1,1,1
      CLS
RETURN
```

Na řadku 10 si nastavte adresář, ve kterém máte soubor intuition.bmap.

Program odskočí na podprogram adresy, který zjistí důležité adresy okna a obrazovky (30). Potom si do proměnné WBScreen& uloží ukazatel na strukturu obrazovky Workbench (40), vyvolá podprogram start, který otevře novou obrazovku a okno (50), funkce na řádku 60 přemístí obrazovku Workbench do popředí. Smyčka FOR-NEXT (70-90) pomalu přesouvá obrazovku WB pomocí funkce MoveScreen() dolů. Dále program znovu zjistí adresy aktuální obrazovky a aktuálního okna a nastaví jejich názvy (110-130). Další smyčka dvacetkrát přesune obrazovku dolů a nahoru, čímž vznikne efekt "klepání" (140-170), okno se zaplní hvězdičkami (180-200) a "popoleze" (210-230). Na řadcích 250-270 se příkazem POKEL změní BitPlane č.2 (číslování od nuly) tak, aby na obrazovce vznikly pruhy. Potom se obrazovka WB znovu vysune nahoru a zakryje obrazovku AmigaBasic (280-300), která se uzavře a program skončí. ■

JAZYK C

2. část

Tomáš Daníček

2.2 PROMĚNNÉ A ARITMETIKA

Následující program vytiskne tabulku funkce f(x) = x * x / 3 - 14.

```
/* Tisk tabulky funkce f(x) = x * x / 3 - 14 pro x = 0, 1, 2 .. 100 */
int Di, Hi, Krok;
main()
{
   float x, y;
   Di = 0; Hi = 100; Krok = 1;
   x = Di;
   while(x < = Hi)
   {
      y = x * x / 3. - 14.;
      printf(" % 5.f \ t % 15.5 f \ n", x, y);
      x = x + Krok;
   }
}
```

V první řádce programu je komentář, který v tomto případě ve stručnosti vysvětluje činnost programu. Všechny znaky mezi /* a */ jsou překladačem ignorovány.

V jazyku C musíme všechny proměnné deklarovat ještě před prvním použitím, obvykle na začátku funkce před prvním výkonným příkazem. Překladač bude hlásit chybu, kdykoliv zapomenete nějakou proměnnou deklarovat. Deklarace se skládá z určení typu a seznamu proměnných.

```
int Di, Hi, Krok; float x, y;
```

Typ **int** znamená, že všechny uvedené proměnné budou celočíselné (integer) a **float** znamená, že se jedná o proměnné s pohyblivou desetinnou čárkou (floating-point). Přesnost a rozsah obou typů je závislý na použitém počítači. Na Amize je **int** 16-ti bitové číslo se znaménkem a leží v rozsahu -32768 a 32767. Číslo typu **float** je 32-ti bitové, má 7 významných číslic a leží v rozsahu 10^{-38} a 10^{+38}. Další z typů proměnných v jazyce C jsou např.:

char	znak - jeden byte
short	krátké celé číslo
long	dlouhé celé číslo
unsigned	celé číslo bez znaménka
double	číslo s pohyblivou čárkou a dvojitou přesností

Základními typy jsou rovněž struktury, uniony, pole, ukazatele na ně a funkce, které s nimi pracují. Se všemi těmito typy se postupně shledáme v dalších kapitolách. Výpočet tabulky funkce začínáme přiřazením

```
Di = 0; Hi = 100; Krok = 1; x = Di;
```

které nastaví počáteční hodnoty proměnných. Pozor! Jednotlivé příkazy musí být odděleny středníkem. Protože dosazujeme do funkce za x hodnoty od 0 do 100, můžeme použít cyklus, který je definován příkazem **while:**

```
while (x < = Hi)
   {
      ...
   }
```

Podmínka v kulatých závorkách je vyhodnocena. Jestliže je pravdivá, tělo cyklu (tj. příkazy ohraničené složenými závorkami) je vykonáno. Potom je podmínka znovu vyhodnocena, a pokud je opět pravdivá, tělo cyklu je opět vykonáno. Cyklus je ukončen, jestliže je podmínka nepravdivá. Tělo příkazu while může být také tvořeno pouze jediným příkazem, např.:

```
while (i < = k) i + +;
```

V případě více příkazů je nutno je ohraničit složenými závorkami. V programu je vhodné zdůrazňovat logickou strukturu a užívat mezery. Doporučuji nechávat mezery kolem operátorů. Vyberte si i vy svůj způsob psaní zdrojových textů a ten stále používejte. Hlavní činnost našeho programu je vykonána v těle cyklu **while**, kde je vypočítána hodnota funkce a přiřazena proměnné y.

```
y = x * x / 3. - 14.;
```

Píšeme 3. a 14., přestože y je typu **float** a 3 i 14 by byly automaticky zkonvertováno na **float** před dělením resp. odečítáním. Je to vlastně jen otázka stylu, přesto je však lepší psát konstanty typu **float** s desetinnou tečkou i když mají celočíselnou hodnotu. Je tím zdůrazněn jejich typ pro čtenáře programu a zajišťuje se i stejné chápání věcí překladačem. V některé z dalších kapitol uvedu pravidla pro konverzi dat na typ **float.** Nyní si všimněme, že přiřazení x = Di; a podmínka while(x< = Hi) funguje tak, jak očekáváme, tj. že proměnné typu int jsou konvertovány na typ float před uskutečněním operací.

Tento příklad také podrobněji ukazuje, že funkce **printf** je obecně použitelná funkce pro formátovaný výstup. Jejím prvním argumentem je řetězec znaků, který má být zobrazen a znak % určuje, kam mají být další argumenty umístěny a jakým způsobem mají být tisknuty na standartní výstupní zařízení. Např. v příkazu:

```
printf("%5.f\t%15.5f\n", x, y);
```

Specifikace **%5.f** znamená, že číslo typu **float** má být zobrazeno s délkou pěti znaků a nemá mít desetinnou část. **%15.5f** určuje, že další číslo bude v délce 15 znaků s pěti číslicemi za desetinnou čárkou. Některé části specifikace nemusí být uvedeny, např. **%6f** určuje, že číslo má být alespoň 6 znaků dlouhé; **%.2f** požaduje 2 místa za desetinnou čárkou a celkový počet znaků není omezen.

%f specifikuje pouze tisk čísla typu float
%d celé číslo
%l dlouhé celé číslo
%x hexadecimální reprezentace
%c znak
%s řetězec znaků
%% zobrazí %

Každý znak % v prvním argumentu funkce **printf** je spojen jen s odpovídajícím druhým, resp. třetím argumentem (**%5.f** je spojen s x resp. **%15.5** s y). Specifikace musí být přesně popsány čísly a typem, jinak dává program nesmyslné výsledky. Připomeňme si, že funkce **printf** není částí jazyka C. Sám jazyk C nemá definovány funkce vstupu a výstupu. Funkce printf je součástí standartní knihovny funkcí.

Příště si ukážeme, jak funguje funkce **scanf**, což je obdoba funkce **printf** pro vstupní operace.

Úkol: Upravte program tak, aby vytiskl záhlaví tabulky. ■

V první části jsme si jen velice stručně řekli, jak je principiálně postavena A500. Nyní bych navázal na obr.2 z minulého čísla. Ze všeho nejdříve si přiblížíme základní funkce procesoru 68000 a CIA obvodu 8520.

Procesor 68000.

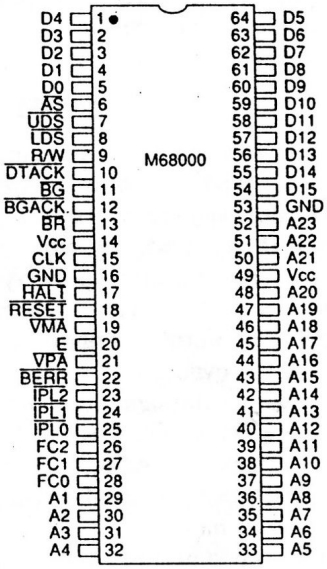
Mikroprocesor 68000 od firmy Motorola je jeden z nejvýkonnějších 32/16 bitových procesorů. Ačkoliv je na trh dodáván od r.1979, objevil se teprve nedávno v různých počítačích v cenové skupině srovnatelné s Amigou. Věřim, že

AMIGA HARDWARE

ČÁST II.

Zdeněk Daníček

nečekáte v této kapitole podrobný popis 68000. Jednak by to nebylo možné, a ani to není cílem tohoto cyklu. Kdo chce vědět více o programování procesoru 68000, počká si na speciální seriál tohoto časopisu, nebo si vyhledá odbornou literaturu. V této kapitole se proto omezíme pouze na popis jednotlivých vývodů a signálních skupin. Základní znalosti o signálech 68000 jsou však nezbytné pro porozumění HW Amigy. Na obr.1 je znázorněn procesor 68000 v pouzdru Dual-In-Line se 64 vývody s označením signálů. Úroveň L je značená čarou nad zkratkou funkce.



Obr. 1.

Obr. 2, jak jsme uvedli, nám dělí vývody do jednotlivých funkčních skupin podle vzájemné příslušnosti. Dále se tedy budeme zabývat jednotlivými skupinami a stručně si vysvětlíme jejich funkci.

Napájení: Vcc a GND

Procesor 68000 pracuje s jedním napájecím napětím +5 V. Vývody pro napájení jsou zdvojeny a umístěny uprostřed (z každé strany jedna dvojice). Důvodem tohoto roymístění jsou krátké přívody, aby nedocházelo ke zbytečným ztrátám.

Hodinový vstup: CLK

68000 potřebuje jen jeden taktovací hodinový kmitočet. Frekvence závisí na zvolené verzi procesoru. U Amigy je použit taktovací kmitočet 7.16 MHz.

Datová sběrnice (Datenbus): D0 - D15

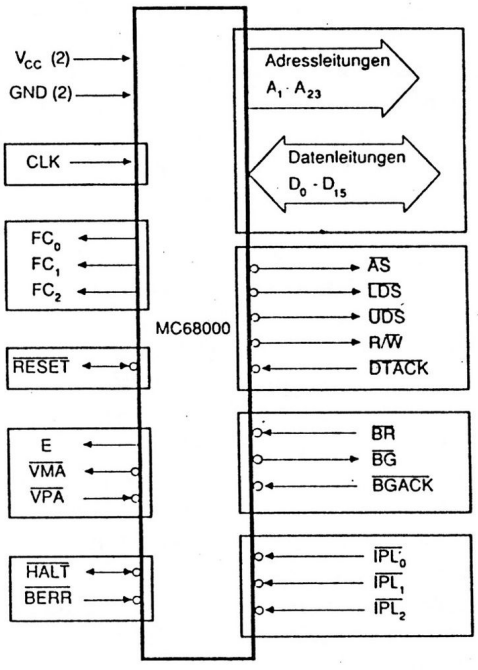
Datová sběrnice je 16-bitová a umí tedy najednou přenášet celé slovo. Při přenosu jednotlivých byte (8 bitů) se podílí na přenos těchto dat pouze polovina sběrnice. Buďto je byte zapisován nebo čten přes vrchních 8 bitů nebo přes dolních 8 bitů.

Adresová sběrnice (Adressbus): A1 - A23

Adresová sběrnice může volat pomocí svých 23 přívodů až 8 Megaslov paměti (223 je 8 Megaslov nebo 16 Mbyte). Vzhledem k tomu, že chybí adresovací bit A0, může tato paměť být adresována pouze v režimu po slovech.

Řídící signály sběrnice v asynchronním modu: AS, LDS, UDS, R/W, DTACK

Procesor 68000 může přistupovat do paměti dvěma různými způsoby. V asynchronním modu signalisuje procesor na vývodu AS (Adress-Strobe tj. adresa platná), že na adresové sběrnici (A1-A23) je k disposici platná adresa. Současně procesor také vydává pokyn na vývodu R/W (Read-Write tj. číst-psát), zda má být na stanovené adrese byte/slovo přečten nebo zapsán. Volbu, zda byte nebo slovo, určují přívody UDS a LDS (Upper-,Lower Data Strobe tj. vrchní-, dolní polovina sběrnice). Vzhledem k tomu, že paměť je vždy adresována po slovech, přenáší procesor při jedno byte-ovém přístupu jen vrchních nebo dolních 8 bitů přes datovou sběrnici. To je signalizováno pomocí UDS a LDS. Pracuje-li procesor s celým slovem (16 bitů), pak vyšle na oba vývody 0. Chce-li naopak jen jeden byte, pak buď na UDS nebo LDS vyšle 0. Druhý zůstává v 1. Jakmile procesor ohlásil požadovaný přístup (na vodičích AS, R/W, UDS a LDS), čeká, dokud mu paměť zpětně nesdělí, že požadovaná data jsou připravena. K tomu slouží vývod DTACK, který přejde do 0, jakmile jsou data připravena. Zapisuje-li procesor data, pak mu paměť přes vodič DTACK sdělí, že data byla převzata.



Obr. 2.: Zde je znázorněno rozložení vývodů do jednotlivých funkčních skupin. Šipky znázorňují směr signálu.

V asynchronním modu se vždy procesor přizpůsobuje rychlosti paměti. Jednotlivá slova a bytes (byty) jsou umístěna v paměti následovně:

Adresa:		zúčastněné dat. vodiče D8-15 UDS = 0	D0-7 LDS = 0
0	slovo 0	byte 0	byte 1
2	slovo 1	byte 2	byte 3
4	slovo 2	byte 4	byte 5
6	slovo 3	byte 6	byte 7
...	.....	.....	.....

Řídící signály sběrnice v synchronním modu: E, VPA, VMA

Abychom lépe pochopili smysl těchto signálů, musíme vyjít ze znalosti situace v době uvedení procesoru na trh. Tehdy totiž nebyl na trhu k disposici žádný periferní obvod pro procesor 68000. Předchozí čip od Motoroly, který byl určen pro serii 6800 (od níž také pochází známý 8 bitový procesor 6502), neznal bez přídavných zařízení možnost asynchronního řízení sběrnice. Proto se objevil u 68000 ještě i synchronní mód zběrnice, tak jak jej známe u 8-bitových procesorů typu 6800 nebo 6502. Na výstupu E je trvale vysílán takt (hodiny) procesoru dělený 10, u Amigy tedy 716 kHz, který slouží jako hodiny pro periferní obvody. Přepnutí ze synchronního módu do asynchronního se provádí přes vstup VPA (Valid Peripherial Adress tj. platná periferní adresa). Tento vstup musí přejít pomocí externího adresového dekodéru do 0, jakmile tuto adresu periferní obvod rozpozná. Procesor na to odpoví vysláním 0 na výstup VMA (Valid Memory Adress tj. platná adresa paměti). Příslušný periferní obvod musí nyní v rámci jednoho hodinového cyklu z E převzít nebo vyslat data. Potom 68000 automaticky opustí synchronní režim, dokud signál VPA není znovu aktivován. To znamená, že periferní obvod je přinucen v synchronním módu v přesně stanoveném okamžiku data převzít nebo vydat.

Systémové řídící signály: RESET, HALT, BERR

Důležitou roli hraje signál RESET, který vlastně znovu odstartuje systém, a to tak, že všechny komponenty se nastaví do výchozího stavu a zahájí se běh programu na pevně dané adrese. Aby se dosáhlo resetu systému, musí se u 68000 stejně HALT jako RESET převést do 0. Jakmile tento vývod znovu přejde do 1, zahájí 68000 běh programu od adresy, kterou nalezne v paměti na adrese 4. Vodič RESET je možné u 68000 zavést na 0 a tím provést inicializaci systému, aniž by došlo ke změně stavu procesoru. S pomocí vodiče BERR (BUS-Error tj.chyba sběrnice) můžeme externě sledovat obvody pomocí procesoru. Např. někde došlo k defektu HW nebo byl učiněn pokus o přístup procesoru na neexistující adresu. Objeví-li se signál BERR, pak 68000 skočí do speciální rutiny operačního systému, která pak převezme další ošetření chyby (GURU-Meditace). Vyskytne-li se během tohoto ošetření chyby nová chyba na sběrnici, zastaví 68000 běh programu a dá signál HALT do 0. Tento stav, tzv. dvojchyba na sběrnici, je jediný případ, kdy se 68000 "zhroutí", což má za následek zastavení běhu programu. Při všech ostatních případech (jednotlivé chyby) skočí procesor pomocí speciálních vektorů do programové rutiny, která pak převezme ošetření chyby a umožní pak další práci systému. Ošetření chyby se nazývá u Amigy "Guru meditace". I zde platí pravidlo, že systém se zboří vždy ve chvíli, kdy zpracovává pro nás ta nejdůležitější data, která dosud nebyla uložena. Jestliže procesor zastavil svoji činnost v důsledku dvojchyby, pak nezbývá bohužel nic jiného, než znovu odstartovat systém pomocí resetu (HALT a RESET do 0). Další funkce vodiče HALT je zastavení procesoru. Přijde-li HALT do 0, ukončí 68000 ještě aktuální přístup do paměti a pak čeká, dokud se HALT nevrátí do 1. Existují ještě další důležité detaily ve spolupráci signálů BERR a HALT, o kterých se zde však nebudeme podrobně zmiňovat.

Provozní stav procesoru: FC0, FC1, FC2

Vodiče FC0 - FC2 signalizují provozní stav procesoru. Jsou možné následující stavy:

FC2	FC1	FC0	Stav
0	0	1	Přístup na uživatelská data
0	1	0	Přístup na uživatelský program
1	0	1	Přístup na řídící data
1	1	0	Přístup na řídící program
1	1	1	Signalizace platného Interruptu

Procesor může pracovat ve dvou režimech, a to v uživatelském módu, nebo v řídícím módu. Program, který běží v řídícím módu, má neomezený přístup do všech procesorových registrů. Operační systém například pracuje vždy v řídícím módu. V uživatelském módu jsou některé registry procesoru pro program nepřístupné. Podrobnosti je třeba hledat v odborné literatuře, zabývající se procesorem 68000. Tři přívody FCx dovolují tedy systémovému HW rozeznat aktuální stav procesoru a patřičným způsobem reagovat. Tak je možné např. v uživatelském módu při přístupu do paměťového obvodu operačního systému vytvořit chybu na sběrnici bus-error (BERR = 0).

Vstupy pro přerušení (interrupt): IPL0, IPL1, IPL2

Signály na třech vstupech pro interrupt (IPL = Interrupt Pending Level) vytváří u 68000 tříbitové binární číslo. Procesor tedy rozezná 8 různých signálů pro interrupt v tzv. rovinách přerušení, přičemž 0 znamená, že není vyžadováno žádné přerušení, zatímco 7 znamená interrupt nejvyšší priority. Každé ze 7 rovin přerušení je přiřazen vektor, který obsahuje adresu rutiny, na kterou při přerušení program odskakuje. Je-li interrupt odpovídající úrovně povolen, pak procesor vyšle na všechny vodiče FCx stav 1 a tím signalizuje, že přerušení rozeznal a že nyní čeká na potvrzení od volající strany. To může následovat přes VPA nebo DTACK. Při potvrzení přes VPA následuje tzv. autovektorové přerušení, to znamená, že procesor skočí na adresu, kterou nalezl podle vektoru příslušné roviny přerušení. Může tedy skočit na 7 různých adres, neboť existuje 7 platných rovin přerušení (rovina 0 značí, že přerušení se nevyžaduje). Protože u systému existuje jen 7 různých zdrojů přerušení, není nutné SW oddělení jednotlivých zdrojů. Jednoduše je každé rovině přerušení přiřazen jeden zdroj přerušení a procesor pak skočí na odpovídající program. Amiga používá jen toto autovektorové přerušení. Mnohem více možností nabízí pro HW rozdělení různých interrupt-údajů tzv. ne-autovektorové přerušení. Protože se u Amigy nepoužívá, zmíníme se o něm jen krátce. Potvrzení přichází přes DTACK a obvod, který intrrupt vyvolal, musí vyslat na datovou sběrnici vektor přerušení, což pak dovoluje pomocí HW obsloužit až 192 různých přerušení.

Signály pro přiřazení sběrnice: BR, BG, BGACK

Tyto tři signály dovolují některému jinému obvodu převzít sběrnici. Toto je využito např. v řadiči harddisku, který data z disku zapisuje přímo do paměti (tzv. DMA t.j. Direct Memory Access, přímý vstup do paměti). Také tyto signály nejsou u Amigy využity. DMA je zde realizován jiným způsobem, který bude vysvětlen příště.

CIA obvod 8520 (Complex Interface Adapter).

Jak je patrno z obr.2 v minulém čísle (1 / 90), obsahuje Amiga 500 dva CIA obvody typu 8520. V úvodu si vysvětlíme, jak jsou uvedené obvody začleněny v systému Amigy. Základní adresa prvního 8520, krátce označen 8520A, je $BFE001. Jeho registry nejdou v adresovém prostoru za sebou, ale s odstupem po 256 bytes (bytech). To znamená,že všechny registry 8520A leží na lichých adresách, protože 8520A je spojen s datovou sběrnicí procesoru pomocí dolních 8 vodičů (D0 - D7). Následující tabulka udává adresy jednotlivých registrů s popisem jejich funkce v Amize (podrobnosti o významu jednotlivých bitů portu budou vysvětleny později).

CIA-A Adresy registrů.

Adresa	Název	D7	D6	D5	D4	D3	D2	D1	D0
$BFE001	PRA	/FIR1	/FIR0	/RDY	/TK0	/VPR0	/CHNG	/LED	/OVL
$BFE101	PRB	Paralelní port Centronics							
$BFE201	DDRA	0	0	0	0	0	0	1	1
$BFE301	DDRB	podle použití buď Vstup nebo Výstup							
$BFE401	TALO	Časovač A pro operační systém při komunikaci							
$BFE501	TAHI	použit klávesnicí							
$BFE601	TBLO	Časovač B je používán z B pro různé úlohy							
$BFE701	TBHI								
$BFE801	E.LSB	Event-counter v CIA-A počítá 50 Hz							
$BFE901	E.8-15	impulsy z síť. napaječe (tzv.Ticks), které jsou odvozeny od síťového kmitočtu							
$BFEA01	E.MSB								
$BFEB01	SP	příjem kódu kláves z klávesnice							
$BFEC01	ICR	intrrupt řídící registr							
$BFEE01	CRA	řídící registr A							
SBFEF01	CRB	řídící registr B							

Druhý 8520 CIA, tzv. CIA-B, leží na základní adrese $BFD000. Jeho registry leží na sudých adresách, protože datová sběrnice CIA-B je spojena s horní polovinou datové sběrnice procesoru (D8-D15). Následuje tabulka adres registrů s popisem jejich funkce.

CIA-B Adresy registrů.

Adresa	Název	D7	D6	D5	D4	D3	D2	D1	D0
$BFD000	PRA	/DTR	/RTS	/CD	/CTS	/DSR	SEL	POUT	BUSY
$BFD100	PRB	/MTR	/SEL3	/SEL2	/SEL1	/SEL0	/SIDE	DIR	/STEP
$BFD200	DDRA	1	1	0	0	0	0	0	0
$BFD300	DDRB	1	1 1	1	1	1	1	1	
$BFD400	TALO	Timer A jen pro seriový přenos dat. Jinak je							
$BFD500	TAHI	nepoužit a je volný.							
$BFD600	TBLO	Timer B slouží k synchronizaci tzv. Blitter s							
$BFD700	TBHI	obrazovkou monitoru. Jinak je nepoužit a je volný.							
$BFD800	E.LSB	Event-counter v CIA-B počítá vodorovné							
$BFD900	E.8-15	synchronizační impulsy. V normálním případě to							
$BFDA00	E.MSB	je 15625 imp. za sec.							
$BFDB00	SP	Nepoužit							
$BFDC00	ICR	interrupt řídící registr							
$BFDE00	CRA	řídící registr A							
$BFDF00	CRB	řídící registr B							

Adresy $BFD000 pro CIA-B a $BFE001 pro CIA-A jsou adresy udávané od firmy Comodore jako základní adresy CIA obvodů. Při podrobnějším prohlédnutí schematu zapojení si jistě odvodíte, že oba CIA se adresují ve stejném rozsahu od A0xxxx do BFxxxx. Volba mezi oběma CIA se provádí na adresovacích vodičích A12 a A13. CIA-A je zvolen, jestliže A12 = 0. CIA-B je zvolen, když A13 = 0. Proto se adresy pohybují mezi A0xxxx a BFxxxx. Vzhledem k tomu, že datová sběrnice CIA-A je spojena s datovou sběrnicí procesoru na vodičích D0 - D7 (liché adresy) a CIA-B pak s D8 - D15 (sudé adresy), je možné současně přistupovat jedním přístupovým slovem na oba CIA současně a to když A12 a A13 je současně na 0. Povel MOVE.W $BF0000 D0 natáhne PA registry obou CIA do D0, přičemž spodních 8 bitů v D0 pak obsahuje PA registr z CIA-A a bity 9-15 u D0 má obsah PA registru z CIA-B. Můžeme si sestavit obecné schéma pro adresování CIA. CIA-A je volán při následující adrese (binárně):

101x xxxx xxx0 rrrr xxxx xxx1

a CIA-B

101x xxxx xx0x rrrr xxxx xxx0

Čtyři písmena r označují bity, které volí požadovaný registr v CIA. Ale pozor! Je možné, že u nových modelů Amigy dojde ke změnám. Pro jistotu doporučujeme ověřit si adresy. ■

(pokračování příště)

HOLD AND MODIFY

Filip Železný

To, co tak proslavilo počítače řady Amiga, je především jejich skvělá grafika. Zajímavé však je, že udávané grafické možnosti těchto počítačů se liší článek od článku, ústa od úst.

Jakou má Amiga rozlišovací schopnost? Kolik dokáže najednou zobrazit barev? 16, 32 nebo dokonce 4096?

To jsou témata častých sporů mezi příznivci počítačů Amiga a počítačů konkurenčních.

Pro úplné pochopení těchto problémů je zřejmě nejlepší se blíže seznámit se způsobem zpracování grafiky hardwarem.

I. Zobrazovací prvky

Na zobrazování se největší měrou podílí několik jednotek hardwaru.

1. Zákaznický čip DENISE

Jeho hlavní úkol spočívá v tom, že kopíruje obraz a jiné prvky grafiky (sprites atd.) z paměti RAM na obrazovku monitoru tak, jak je určeno.

2. Čip pro rychlý přenos dat - BLITTER

Bez této součásti by se Amiga se svým "oknovým" operačním systémem snad ani neobešla. Blitter totiž dokáže velmi rychle přenášet a kopírovat pravoúhlé výřezy obrazu. Jeho rychlost je asi 14 000 000 bodů za sekundu, to znamená, že by dokázal celou standardní Workbench-obrazovku přemístit rychleji než za jednu padesátinu sekundy.

3. Koprocesor COPPER

Copper je součástí zákaznického čipu FAT AGNUS a používá se pro synchronizaci operačního systému a zobrazování na monitoru.

Máme-li např. na monitoru několik systémových obrazovek (screens), pak na rozhraní mezi nimi zajišťuje Copper změnu grafického módu, změnu barev atd. Copper pracuje v závislosti na aktuální poloze zobrazovacího elektronového paprsku na monitoru (popsáno dále), a protože je to samostatný procesor, má také vlastní instrukční soubor a vlastní program uložený v paměti RAM. Jeho instrukční soubor je však velmi jednoduchý. Zahrnuje pouze následující instrukce:

a) Pro zápis do paměti - MOVE (hodnota,adresa)

Tato instrukce je šestnáctibitová a lze s ní zapisovat pouze do paměťové oblasti hardware - registrů zákaznických čipů, které jsou v paměti umístěny od adresy $DFF000. Touto instrukcí je však možno i provádět skoky v programu. (Zápisem do PC-registru Copperu.)

b) instrukce WAIT (x,y)

Tato instrukce vyčkává, dokud se zobrazovací paprsek nenachází v pozici (x,y).

c) instr. pro jednoduchou podmínku - SKIP (x,y)

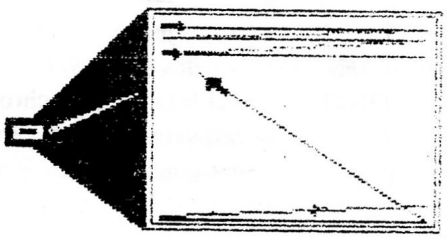
Instrukce přeskočí instrukci následující, dosáhl - li již el. paprsek uvedených souřadnic.

I přes velmi jednoduché možnosti Copperu je tento procesor opravdu významným pomocníkem systému. Editaci programů umožňuje kromě jiného i tzv. CopperMonitor.

Na zobrazování grafiky se samozřejmě podílí i hlavní procesor Motorola 68000 tím, že modifikuje obraz uložený v paměti tak, že zasahuje do hardware - registrů, ovládá ostatní čipy atd.

II. Jak zobrazuje monitor.

Obraz se na monitor nekopíruje jako celek. Na obrazovce se velkou rychlostí pohybuje elektronový paprsek, který rozsvěcuje jednotlivé body. Schéma pohybu je zakresleno na obrázku.



Popis: Paprsek se pohybuje zleva doprava a vykresluje body na horizontální řádce. Na konci řádky se začne nerozsvícený vracet nazpátek, ale o řádek níž. To provádí až do konce obrazovky. Je-li obraz dokreslen, vrátí se paprsek zpátky do výchozí polohy. Standardně se obraz takto vykreslí za sekundu 50 (60) krát.

Každý bod můžeme rozdělit na tři složky: červenou, zelenou a modrou (Red, Green, Blue)

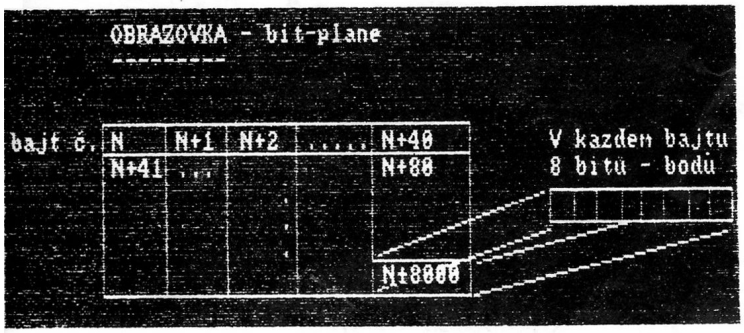
Podle toho, jak intenzivně jsou jednotlivé složky rozsvíceny, vznikají jejich složením různé barvy.

U každé z těchto složek dokáže Amiga nastavit intenzitu

jasu v šestnácti stupních. Z toho také vyplývá omezení 4096 barev. (16 * 16 * 16 = 4096)

III. Uložení obrazu v paměti RAM.

Jak je též uvedeno v článku pojednávajícím o použití knihoven v Basicu, je každý bod na obrazovce monitoru zaznamenán v paměti jako jeden bit způsobem, který ukazuje ob[illegible]ek.



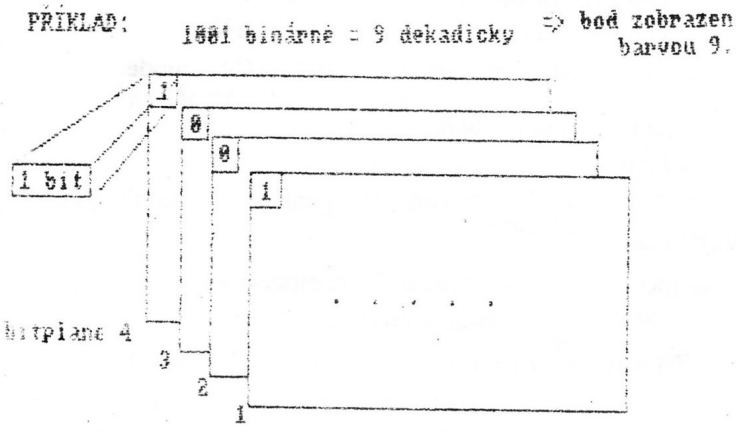
Poznámky:

Matici bitů zakreslené na obrázku se říká BitPlane.

- Je-li bit nastaven na hodnotu 1, je bod "rozsvícen" a naopak.

- Uvědomte si, že bajt v paměti obsahuje 8 bitů, tedy zaznamená osm bodů.

Vícebarevný obrázek se řeší "překrytím" několika BitPlanes. Jeden bod je prezentován jako jeden bit, který je však zaznamenán se stejnými souřadnicemi ve všech použitých BitPlanes, ale s různými hodnotami. Kombinace těchto hodnot udává číslo barvy, kterou bude bod zobrazen.



IV. Grafické módy

- K zobrazování je možno použít dvojího HORIZONTÁLNÍHO rozlišení:

- 320 bodů na řádce (LoRes)

- 640 bodů na řádce (HiRes)

Při LoRes módu se jeden bod vykresluje po dobu dvou taktovacích cyklů čipu Denise a je zde možno použít 5 BitPlanes. (32 barev)

Při módu HiRes trvá vykreslení bodu jeden taktovací cyklus a je možno použít pouze 4 BitPlanes. (16 barev)

- Existují dva způsoby VERTIKÁLNÍHO rozlišení:

- INTERLACED

- NON-INTERLACED

Při módu non-interlaced je na obrazovce vykresleno 200 resp. 256 řádek bodů, podle toho, zda Amiga pracuje v normě NTSC nebo PAL. Rozdíl mezi nimi není ve velikosti bodů, ale v tom, že se v normě NTSC po řádku 200 body nevykreslují a obraz je tedy menší.

Při módu interlaced se výška všech bodů dvakrát zmenší - tedy počet řádek se dvakrát zvětší. Pří zobrazování jou vykresleny nejdříve všechny liché řádky a potom řádky sudé. To znamená, že celkově je obraz vykreslen dvakrát pomaleji než normálně. To způsobuje viditelné blikání.

V tomto módu lze dosáhnout vertikálního rozlišení až 512 řádek.

Při zobrazování se vždy vynechává určitý počet bodů od okrajů obrazovky. Zvláštní je mod OVERSCAN, který tyto body aktivně využívá, čímž se rozlišitelnost ještě zvýší. Body na okrajích jsou v tomto případě zkreslené, což je způsobeno mírným zaoblením obrazovky.

Dalším zvláštním módem je tzv. DUAL PLAYFIELD mode. Při něm jsou vytvořeny dvě nezávislé obrazovky za sebou. S každou je možno pohybovat a každá může obsahovat nejvýše tři BitPlanes - tedy 8 barev, přičemž barva č. 0 je tzv. transparentní - průhledná.

Tohoto módu je možno využít např. pro hry, kdy je potřeba nezávislá krajina a před ní ovládací panel, nebo pro vytváření trojrozměrných efektů (např. hra Shadow Of The Beast).

Při tzv. EXTRA HALFBRITE modu (EHB) se může zobrazit 64 barev najednou, což je dosaženo snížením jasu na polovinu.

V poslední době kdosi vytvořil nový mód HCS, při kterém je údajně možné použít cca 25000 barev najednou.

Mezi širokou veřejností je zřejmě nejvíce diskutovaný velmi výkonný mód HOLD AND MODIFY (HAM), pomocí něhož je najednou zobrazitelných až 4096 barev. Jsou zde však určitá omezení.

Zobrazování při Hold And Modify módu je poněkud odlišné od normálního. Informace o barvě bodu jsou uloženy následující technikou:

- Pokud jsou bity v BitPlane č.6 a č.5 nastaveny oba na 0, pak číslo barvy určují příslušné bity v BitPlane č. 1 až 4.

- Jestliže jsou bity v BitPlane č.6 a č.5 nastaveny na 0 a 1, převezme se barva z předešlého bodu (HOLD) a změní se pouze barvová složka BLUE, jejíž hodnotu určují bity v BitPlane č.1 až 4 (MODIFY).

- Jsou-li bity v BitPlane č.6 a č.5 nastaveny na 1 a 0, převezme se barva z předešlého bodu a změní se pouze barvová složka RED, jejíž hodnotu určují bity v BitPlane č.1 až 4.

- V případě, že jsou bity v BitPlane č.6 a č:5 nastaveny oba na 1, převezme se barva z předešlého bodu a změní se pouze barvová složka GREEN, jejíž hodnotu určují bity v BitPlane č.1 až 4.

Hold And Modify mód se nedá použít v kombinaci s módem HiRes, tuto nevýhodu však některé nové programy odstraňují (Digi Wiew 4.0 apod.).

V tomto módu se jedinečně provádí např. stínování, ale již zmíněné omezení tkví v tom, že na přechodu mezi barvami lze pozměnit pouze jednu barvovou složku, to znamená, že by neměly být vedle sebe zobrazitelné dva body s více složkami rozdílnými (např. černá - bílá). To se odstraňuje tak, že místo modifikování jednotlivých složek se na přechodu použije standardní, předem nadefinovaná barva (tedy bity v BitPlanes č.6 a č.5 mají oba hodnotu 0), kterou nazvěme přechodová. Bity, které určují číslo přechodové barvy, jsou čtyři (BitPlane 1 až 4), tedy použitelných přechodových barev je šestnáct. Pokud je na obrázku více rozdílných přechodů než šestnáct, ani pak tím teoreticky naše možnosti nekončí, protože i přechodová standardní barva se dá těsně před použitím (v průběhu zobrazování) modifikovat např. pomocí Copperu (nechceme-li tím zaměstnávat hlavní procesor). Mějte však na vědomí, že Copperu trvá osm taktovacích cyklů (tedy dobu vykreslení osmi LoRes-bodů), než změní šestnáctibitový registr v paměti. Tedy část Copper-programu, modifikující přechodovou barvu, bude mít tuto strukturu:

```
.
.
WAIT (X,Y-8)
    MOVE COL,$DFF180 ;reg. barvy č.0
.
.
.
```

(X,Y znamenají souřadnice prvého (levého) stykového bodu, COL znamená barvu pravého stykového bodu, jako přechodová je použita standardní barva č.0)

Použití HAM je tedy, jak vidět, velice jednoduché.

Složitou optimalizaci barevného zpracování Vám obstarají příslušné grafické programy (Digi Paint, Digi View, Photon Paint, Pix Mate atd.), z nichž ty nejmodernější jsou dokonce i velmi rychlé.

Přes všechny potíže se zpracováním je mód HAM bezesporu velmi výkonný a prakticky záleží pouze na zdatnosti programátora, do jaké míry se mu podaří odstranit jistá omezení.

Následující program v jazyku C vytvoří obrazovku (screen) v modu Hold And Modify:

```
#include /intuition.h
#include /screens.h
        /* Pro případné zásahy do struktury obrazovky */
#include graphics/gfx.h
        /* Pro případné použití BitMap struktury
         ( Tato struktura obsahuje mimo jiné i adresy
        všech BitPlanes a ty jsou pro mod HAM důležité)*/
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase; struct Screen *scr;
        /* Deklarace systemových knihoven a obrazovky
         a jejich paměťových ukazatelů */
struct NewScreen ns = {
    0,0,        /* X,Y lev. hor. rohu obrazovky  */
    320,200,    /* šířka a výška */
    1,          /* počet bitplanes */
    0,1,        /* detailní a bloková barva */
    HAM,        /* DISPLAY - mody */
    CUSTOMSCREEN,  /* typ obrazovky */
    NULL,       /* ukazatel fontu */
    "Hold And Modify Screen",   /* název */
     NULL,      /*ukazatel na výpis 'gadgets' */
    NULL        /* ukazatel na vlastní BitMap - možno použít */
};
main()
{
    if ((IntuitionBase = OpenLibrary ("intuition.library",0L))
= = NULL)
    {
        printf ("INTUITION - knihovna nejde otevřít.");
        exit();
        }   /* Otevření knihovny INTUITION a kontrola */
    if ((scr = OpenScreen (&ns)) = = NULL)
        {
        printf ("Obrazovka nejde otevřít");
        CloseLibrary (IntuitionBase);
        exit();
        }       /* Otevření obrazovky a kontrola */
}
```

Na závěr shrneme možné grafické mody:
Vzhledem k rozlišitelnosti:

NTSC PAL	
320 x 200 (256)	- non-interlace,LoRes
320 x 400 (512)	- interlace,LoRes
640 x 200 (256)	- non-interlace,HiRes
640 x 400 (512)	- interlace,Hires
přes celou obrazovku	- OVERSCAN mode

Vzhledem k počtu barev:
1,2,4,8,16,
u LoRes i 32 - Normal Playfield mode (podle počtu BitPlanes)

64	- Extra Halfbrite mode
4096	- Hold And Modify mode

Zvláštní módy jsou: Dual Playfield mode - dvě nezávislé obrazovky za sebou a Sprites - na obrazovce je možno zobrazit tzv. sprites (malé grafické objekty, určené pro animaci). ■

HRY

Tomáš Adamec

International Karate +

■ Archer MacLean 1988

TEST 2 / 90		
Grafika	84 %	
Zvuk	70 %	Celkem
Idea	72 %	76 %
Přitažlivost	77 %	

IK+ je bojová hra ve stylu karate shotokan. Obsazením úderů vás asi příliš nepřekvapí - až na hlavičku a roznožku jsou bojovníkovy možnosti stejné jako u tuctu obdobných her. IK+ se však přeci jenom něčím od ostatních karate-her lišj. Autoři totiž přidali k obvyklé dvojici bojovníků bojovníka třetího, čímž dostala hra punc originality. IK+ umožňuje i hru dvou hráčů (F2), kde třetí bojovník zůstává pod kontrolou počítače.

Tím je umožněn boj "všech proti všem". Častěji však asi dojde k alianci obou hráčů proti stále nebezpečnějšímu karatistovi-počítači. IK+ obsahuje dvě pěkně provedené mezihry, které se pravidelně střídají. Můžete též měnit rychlost pohybu zápasníků (klávesy F6-F10). Autoři projevili i smysl pro humor - zkuste někdy během boje stisknout klávesu "T". Můžete též přepínat mezi němčinou a angličtinou a využívat další volby klávesnice. Např.: Mezerník - pauza, S - změna stínování, B - změna štítu v první mezihře, Help - barva pásku... Tyto možnosti (a některé další) jsou popsány na informačním panelu, který se zobrazí asi po půl minutě čekání v úvodu hry. ■

TAD

Výborný design, grafika se smyslem pro detail (padající listí, pavouci, ryba, periskop apod.). Hudba je pěkná, zvukové efekty slušné. Vysoká míra hratelnosti. Celkový nápad dobrý, provedení výborné.

TOMÁŠ

IK+ je velice dobře provedené karate. Hra má jeden dobrý nápad - tři bojovníky, čímž je umožněno, aby dva hráči spojili své síly proti počítači. IK+ obsahuje velké množství kopů, krytů a úderů. Grafika i zvuk jsou na dobré úrovni.

Continental Circus

■ Taito 1989

TEST 2 / 90		
Grafika	68 %	
Zvuk	66 %	Celkem
Idea	35 %	58 %
Přitažlivost	60 %	

Jednoduché zpracování závodů formule 1. Startujete na různých okruzích v mnoha zemích světa a postoupit na další trať můžete jedině po dokončení předešlého okruhu. Ten však musíte dojet ve stanoveném časovém limitu a navíc musíte dosáhnout stanoveného umístění (nebo lepšího, samozřejmě). Váš závodní vůz je pouze dvourychlostní a řídit jej není vůbec složité. Důležité však je, abyste se dokázali vyhnout ostatním závodníkům. V případě srážky s jiným vozem začne váš stroj hořet a vy musíte rychle zajet do boxu, kde bude automobil uhašen. Jinak dojde k výbuchu a ztrátě života. ■

TAD

Velice rychlá nenáročná hra s průměrnou grafikou i zvukem. Brzy mi ale začala vadit absence jakéhokoliv originálního nápadu, který by dokázal CONTINENTAL CIRCUS nějak odlišit od spousty her podobného ražení. Nahrávací sekvence na začátku hry je slabá. Zkrátka - žádná sláva.

TOMÁŠ

CONTINENTAL CIRCUS je pokusem o ztvárnění závodů formule 1. Ovládání závodního vozu je však nepřesné, grafika je na nízké úrovni, zvukové efekty nejsou příliš věrné. Celkově se dá říci, že CONTINENTAL CIRCUS patří mezi nejhorší hry svého druhu.

Silkworm

■ Tecmo 1988

TEST 2 / 90		
Grafika	80 %	
Zvuk	69 %	Celkem
Idea	47 %	72%
Přitažlivost	90 %	

Děj většiny střílaček (pokud zrovna neovládáte nějakého supermana) je umístěn do vesmíru. Hráč řídí jakousi po zuby ozbrojenou kosmickou loď a prolétává vesmírnými městy, stanicemi a jinými podivnými konstrukcemi. Přitom bojuje s různými vetřelci, nepřátelskými koráby apod.

Autoři SILKWORMu sice vytvořili zcela klasickou střílečku, avšak děj situovali na Zemi. SILKWORM umožňuje

hru dvou hráčů, přičemž jeden ovládá vrtulník a druhý vojenský džíp. Namísto vesmírných potvor a kosmických korábů na vás útočí vrtulníky, tanky, obrněné transportéry, letadla, kanóny protiletadlové obrany a další. Podobně jako u většiny ostatních střílečeк můžete sbírat různé prémie (např. zdvojenou střelbu). Hra obsahuje několik úrovní, které se postupně stávají těžšími a těžšími. ■

TAD

SILKWORM se mi vcelku zalíbil. Grafika ovšem není nijak zázračná. Abnormální je počet objektů (nepřátel), kterých občas bývá na obrazovce až překvapivě mnoho. SILKWORM je hra strhující a tato vlastnost spolu s dobrými nápady (např. v grafice pozadí) zaručují, že výsledný dojem je dobrý.

Crack

■ LINEL Switzerland 1988

TEST 2 / 90		
Grafika	71 %	
Zvuk	55 %	Celkem
Idea	39 %	56 %
Přitažlivost	57 %	

Zástupce klasického herního stylu "bourání zdi". Hráč ovládá myší plošinku, která slouží k odrážení kuličky. Při každém střetu kuličky se zdí dojde k vybourání jedné či více cihel. Mluvíme sice o zdi, ale jedná se spíše o většinou velmi rafinovaně tvořené obrazce.

Celý CRACK je situován do pravěku. Za každou zdí je skryt stylový obrázek, který se během bourání obrazce odhaluje. Z většiny duhově zabarvených cihel vypadávají barevná kolečka, která je třeba sbírat - přinášejí totiž různé výhody: Zdvojení kuličky, střelbu, rozšíření či zdvojení plošinky, posun o kolo kupředu, život (plošinku) navíc atd. Zároveň však mohou přinést i nevýhody - obrácení smyslu ovládání, posun o kolo zpět apod. Zařazena je i mezihra, ve které musíte pochytat co možná nejvíce kokosových ořechů, které padají z palem. K dispozici je též mód pro dva hráče a tzv. battlemód. Zabudován je i editor zdí, kde je možné jednoduchým způsobem vytvářet nové zdi, či staré zdi modifikovat. ■

TAD

Jednoduchá hra, zpracovaná na dobré úrovni s použitím několika dobrých nápadů. Avšak již celková koncepce ji řadí mezi sice pěkné, nicméně poněkud nudné hry. Hodí se skutečně jenom pro nenáročný odpočinek.

TOMÁŠ

Když srovnám CRACK s jinými hrami tohoto druhu, skončí nejspíše až na posledním místě. Asi jedinou opravdu dobrou věcí u CRACKu je editor (přímo na základní disketě), kterým si můžeme postavit libovolnou zeď.

R-Type

■ Irem 1987

TEST 2 / 90		
Grafika	73	
Zvuk	71	Celkem
Idea	40	61
Přitažlivost	57	

R-TYPE je klasickou střílečkou s horizontálním posunem obrazu. Hráč ovládá vesmírnou loď, která se musí probít několika sektory plnými nepřátel. Na konci každého sektoru na vás čeká jedna super-potvora, kterou musíte bezpodmínečně zničit. Během cesty přibíráte další a další zbraně, o které však můžete při ztrátě života přijít. Tato hra (a jí podobné) vyžaduje především dobrý postřeh, pevné nervy a joystick s automatickou střelbou! ■

TAD

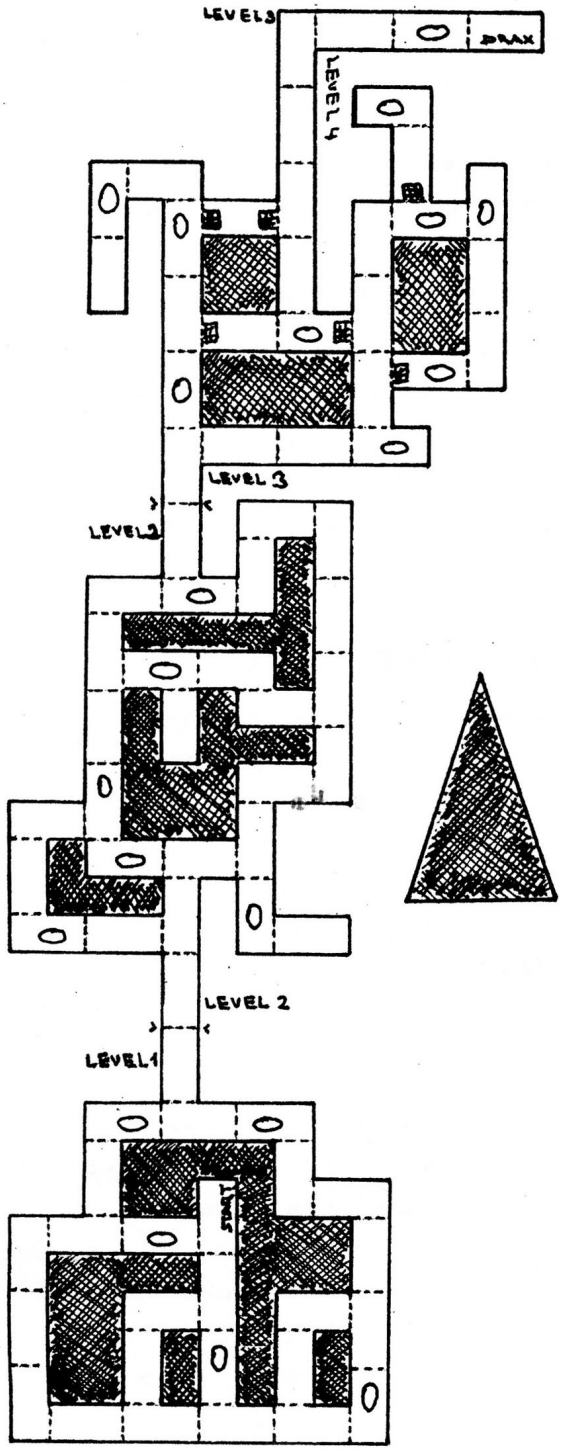
Tato hra mě příliš nepřekvapila - je to vcelku pěkná střílečka, zpracovaná v poměrně dobré grafice a se slušným zvukem (především úvodní hudba je kvalitní). Musím však poznamenat, že R-TYPE se mi zdá být pro obyčejného smrtelníka poněkud nehratelný, zvláště pak ve vyšších úrovních.

TOMÁŠ

Klasická hra, kde hráč střílí opravdu na všechno, co se hýbe. Prostředí, ve kterém se kosmická loď pohybuje, je vcelku dobře nakreslené, stále se měnící krajina a nové nápady zaujmou každého, kdo má tento styl her v oblibě.

DEFENDER OF THE CROWN

Najímání vojáků či převádění vojáků z hradní posádky do armády (při jejich větším množství) po jednom muži je pro hráče únavné, kromě toho dochází ke zbytečnému opotřebování levého tlačítka myši (najmout 200 mužů znamená 200x "kliknout" levým tlačítkem myši). Tomu se lze vyhnout jednoduchou "fintou". Při najímání či převádění vojáků nejprve stiskněte pravé tlačítko myši a držte jej. Nyní normálně stiskněte levé tlačítko myši. Nenajali jste jednoho, nýbrž pět mužů!



Opravenka

Omlouváme se všem čtenářům za chybu, která vznikla při sestavování minulého čísla. U vysvětlivek k mapě hry Obliterator byly opomenuty grafické segmenty, čímž se legenda vlastně stala nesmyslnou. Mělo to vypadat takto:

V	Vetřelci	S	Místo možného startu
Z2	Rychlopalná zbraň	Z3	Raketomet
Z4	Bazooka		Energetický mrak - zničit
·A	Označuje průchod		Energetické pole
	Smrtelné miny		Nejbezpečnější cesta
	Náboje		Sloup. Zničit bazookou

Regenerátor síly, možnost uschování pozice
Slot pro aktivátor
Plazmový komponent pro stroje. Vezměte s sebou.
Hlavní zbraňový komponent. Vzít s sebou.
Počítačový systém - data. Vzít s sebou.
Únikový aktivátor. Vzít s sebou do letounu.
Deaktivátor bojových štítů. Vezměte s sebou.

Barbarian II

■ Palace Software 1989

TEST 2 / 90		
Grafika	84 %	
Zvuk	85 %	Celkem
Idea	75 %	
Přitažlivost	87 %	83 %

V závěru hry BARBARIAN porazil Conan Draxovy bojovníky a osvobodil zakletou princeznu Marianu. Drax však před Conanem utekl a tajnou chodbou se dostal do svého podzemního sídla, kde přísahal pomstu celému království. Pouze dva bojovníci - Conan a Mariana - mohou Draxe zadržet.

Nejprve musíte projít tři úrovně hry (pustinu, jeskyni a sklepení), teprve potom vstoupíte do Draxova tajného útulku. Během cesty se orientujte podle magického meče, který je zobrazen v dolní části obrazovky. Jeho špička vždy míří na sever. Po dosažení Draxova útočiště musíte nejprve porazit Žijící Modlu, Velkého Démona a nakonec samotného Draxe.

Hru začínáte s pěti životy. Pokud si vezmete lebku, získáte jeden život navíc. V každé ze tří úrovní si musíte vzít dva magické předměty, které vám pomohou v postupu dále. Je nezbytně nutné je najít.

SEKERA zvýší vaši sílu.

KOULE vás chrání proti Draxově magii.

MEDICÍNA zvýší vaše mentální schopnosti.

KLÍČ otevírá dveřní mříže.

ŠTÍT vás ochrání proti okamžité smrti po zásahu démonovým ohněm.

KÁMEN slouží k zabití Žijící Modly.

A dvě rady na závěr: V každé ze tří úrovní se nachází jedna obzvláště nepříjemná potvora. V pustině je to drak, v jeskyních lidožravý tygr, ve sklepeních obrovské larvy. Soubojům s těmito netvory se raději vyhněte. Chcete-li projít tunelem či dveřmi, můžete použít joystick. To je však nepřesné a nepohodlné. Mnohem výhodnější je stisknout mezerník. ■

TAD

BARBARIAN II je výborná akční hra, zpracovaná ve velice slušné grafice a s dobrým zvukovým zázemím (především v třetí úrovni). Všechny animované objekty (potvory) jsou zpracovány velice pěkně s velkou dávkou fantazie. Boj se sekerou je dobrým nápadem. Nahrávací sekvence nemá chybu.

F / A - 18 Interceptor

■ Electronic Arts 1988

TEST 2 / 90		
Grafika	82 %	
Zvuk	65 %	Celkem
Idea	82 %	79 %
Přitažlivost	86 %	

Tento program je simulátorem amerického stíhacího letounu F/A-18 Hornet. Ihned po svém uvedení na trh si INTERCEPTOR získal velikou oblibu - nabízel totiž velké zážitky bez nutnosti dlouhodobého tréninku. Dnes vám tedy přinášíme kompletní popis této hry.

Při prvním setkání s INTERCEPTORem jste zajisté ocenili možnosti FREE FLIGHT a PRACTICE MANEUVRES. Nutnost dobré znalosti vašeho letounu se projeví především během vzdušných soubojů. Proto musíte bezpodmínečně zvládnout alespoň základní letecké manévry, jako jsou např. Aileron Roll, Inside Loop, Vertical Half-loop a další.

QUALIFICATION

Před postupem do dalších kol musíte nejprve dokázat svoji zdatnost tím, že vzlétnete z letadlové lodi a opět na ní přistanete. Na první pohled se vám to asi zdá jednoduché. Ale pilot-začátečník brzy zjistí, že tomu tak není. Nováčci se obvykle zachovají tak, že vzlétnou, v určité vzdálenosti od lodi otočí letoun čelem vzad a potom se snaží co možná nejpřesněji nabrat kurs na přistání. My vám nyní předvedeme malou "fintu", "loop-metodu", kterou vynalezl Mike Pattend. Postupujte takto:

Po startu z paluby letadlové lodi naberte výšku do 1500 stop a ve vzdálenosti asi dvou mil od lodi (NESMÍTE však měnit směr!) proveďte halfloop. Nyní se vracíte zpět k lodi, ale hlavou dolů. Minete loď a ve vzdálenosti dvou mil loop dokončete. Ocitnete se v normální poloze s lodí přímo před vámi. Nyní stačí letět ve výšce okolo 150 stop (výška paluby je 125 stop), snížit rychlost na 50%, vysunout podvozek a ostruhu a přistát tak, aby se ostruha zachytila o gumová lana na palubě. Nyní zapněte brzdy pomocí backspace a snižte rychlost na minimum. Přistanete- li dobře, objeví se zpráva "Qualification complete".

Po ukončení kvalifikace můžete začít s plněním jednotlivých poslání.

VISUAL CONFIRMATION

Ze severozápadu se k San Francisku blíží neznámý objekt. Vaším úkolem je provést jeho identifikaci. Vzlétněte z Moffetského letiště a naberte kurs 310. Jakmile se UFO objeví na 40-ti mílovém radaru, umístěte jej zhruba proti sobě a obdržíte informaci o označení, rychlosti a výšce tohoto objektu. Samozřejmě, že se jedná o Mig-29. Obdržíte-li zprávu o identifikaci letounu, můžete se vrátit na základnu. Tím je tento úkol splněn.

EMERGENCY DEFENCE OPERATION

Presidentské letadlo AF-1 se blíží k Moffetskému letišti. Je poškozeno nepřátelskou raketou. Vaším úkolem je za každou cenu zabránit Migům v přiblížení se k prezidentskému letadlu. Zaútočte jako první a co možná nejrychleji sestřelte oba Migy. Pokud se vám podaří Migy zničit, nebo je alespoň zdržet na dobu nutnou k přistání AF-1, splníte úkol a můžete se vrátit na mateřskou loď.

INTERCEPT STOLEN AIRCRAFT

Dva stíhací letouny F-16 byly ukradeny a odlétají nejvyšší možnou rychlostí na severozápad v doprovodu dvou Migů. Je sice možné oba letouny doprovodit zpět, ale jejich sestřelení je podstatně efektivnější. Sestřelte nejprve prvního , potom druhého F-16. Nesmíte vypustit najednou dvě rakety, protože zaměřovací systém by navedl obě střely na stejné letadlo. Po sestřelení obou F-16 se můžete vrátit na základnu. Souboj s Migy již není nutný.

SEARCH AND RESCUE OPERATION

Jakýsi pilot byl sestřelen. Podařilo se mu však sklouznout s poškozeným letadlem na mořskou hladinu. Chcete-li jej zachránit, musíte spěchat, neboť stíhačka příliš dlouho na hladině nevydrží. Musíte na místo nehody dopravit záchranný člun. Naberte kurs na sestřelené letadlo (jeho poloha je zobrazena na mapě při začátku poslání). Je třeba letět dosti nízko (pod 250 stop) a velmi pomalu (asi 30%). Až bude sestřelený pilot pod vámi, vypusťte záchranný člun. Pokud člun odhodíte příliš brzy, musíte začít znovu. Pokud odhodíte člun pozdě, čeká vás totéž. Až úkol splníte, vraťte se na mateřskou loď.

INTERCEPT CRUISE MISSILE

Vaším úkolem je zničit řízenou střelu s plochou dráhou letu, směřující na San Francisko. Raketa bude na mapě vypadat jako velmi rychlý Mig, nebude však měnit směr a poletí velmi nízko. Její zaměření není příliš obtížné. Pak se ale musíte přiblížit k raketě natolik, abyste ji mohli zasáhnout. K tomu je potřeba zapnout přídavné spalování. Jakmile obdržíte zprávu "IN RNG" a uslyšíte zvukový signál, odpalte dvě rakety (pro jistotu).

CARRIER SUB MISSION

Na severozápadě se nachází nepřátelská letadlová loď, ze které byla právě vypuštěna řízená střela na San Francisko. Loď je zabezpečena dvěma Migy. Vaším úkolem je zničit řízenou střelu, sestřelit oba Migy a nakonec potopit nepřátelskou letadlovou loď. K potopení lodi však budete potřebovat VŠECH ŠEST raket. Proto nejprve zničte řízenou střelu a oba Migy, potom se vraťte na základnu, doplňte palivo a munici a vydejde se k nepřátelské lodi. Přibližujte se ve výšce okolo 500 stop. Když se loď objeví na obrazovce, klesněte asi na 150 stop. Všech šest raket musí zasáhnout velitelskou věž nepřátelské letadlové lodi. Váš palubní počítač však odmítne uznat letadlovou loď za normální cíl, proto je třeba zaměřit rakety manuálně. Umístěte věž přesně doprostřed střeleckého kříže, snižte tah motorů asi na 50% a vypusťte tři rakety. Nepokoušejte se vystřelit větší počet ...et, není to v tuto chvíli možné. Nyní se věži vyhněte a proveďte nálet z druhé strany. Pokud všech šest raket zasáhne velitelskou věž, toto vyletí do povětří a loď se potopí. Pokud některá z raket minula svůj cíl, musíte se vrátit na základnu, opět doplnit munici a poté útočnou akci dokončit.

A několik užitečných rad závěrem:

1) Vždy používejte naváděcí systém raket, a to i tehdy, je-li cíl ještě mimo dostřel. Ulehčí vám to orientaci.

2) Při vypuštění rakety je zásah nejjistější v případě, že se cíl nachází v zaměřovacím kříži.

3) Je-li kolem vás větší počet nepřátel, nikdy nestřílejte více jak jednu raketu. S municí musíte šetřit.

4) Neztrácejte čas používáním kanónu. I když se přiblížíte na ...ný dostřel, ukáže se kanón jako zbraň pomalá a neúčinná.

5) Pokud na vás nepřítel vystřelí raketu, můžete ji zneškodnit pomocí odvaděčů. Máte k dispozici dva typy odvaděčů - Flares proti tepelně naváděným a Chaffs proti radarem naváděným raketám.

6) V průběhu leteckého souboje je nutné stále kontrolovat pohyby protivníka. Pokud vám soupeř zmizí z obrazovky, proveďte nějaký nečekaný obrat a jakmile se nepřítel opět objeví ve vašem zorném poli, vypusťte raketu dříve, než stačí Mig zareagovat.

OVLÁDÁNÍ NA KLÁVESNICI

F1 až F10	Tah motorů po 10% (2 x F10 = přídavné spalování)
+	Tah motorů po 1% +
-	Tah motorů po 1% -
Backspace	Brzdy
Esc	Opět spustit zvolenou misi
R	Nastavení dosahu radaru
T	Zapnutí naváděcího systému
P	Pauza
[	Zoom -
]	Zoom +
A	Vysunutí ostruhy
F	Flare - odvaděč tepelně naváděných raket
G	Vysunutí podvozku
H	Zapnutí / vypnutí HUDu
J	Zapnutí obranného systému ECM
Return	Změna zbraní
C	Chaff - odvaděč radarem naváděných raket
M	Mapa
<	Kormidlo doleva
>	Kormidlo doprava
Mezerník	Střelba zvoleným typem zbraně
Kurzor. kl.	Stejná funkce jako u joysticku
SHIFT + E	Katapultace
SHIFT + Esc	Hlavní menu
SHIFT + F	Odhození záchranného člunu (u poslání č. 4)

ČÍSLICOVÝ BLOK (V KABINĚ)

1	Pohled dolů
2	Pohled dozadu
4	Otáčení v kabině doleva
5	Uchová zaujatou pozici
6	Otáčení v kabině doprava
7	Pohled vzhůrú
8	Normální pohled vpřed

Enter - pohled na letadlo ZVENKU

1	Šikmo zleva zezadu
2	Zezadu
3	Šikmo zprava zezadu
4	Zleva
5	Shora
6	Zprava
7	Šikmo zleva zepředu
8	Zepředu
9	Šikmo zprava zepředu
.	Zespoda
SHIFT + Enter	Pohled z řídící věže (podle místa startu buď z letadlové lodi, nebo z letiště).

Pozn.: Všechny směrové údaje berte ze standartní pozice "2", tj. venkovní pohled zezadu. ■

TAD

INTERCEPTOR je jednoduchý simulátor. Zpracován však byl perfektně. Ovládat INTERCEPTOR je snadné, let s ním je požitkem i pro začátečníka. Všechna poslání jsou zajímavá a hlavně není složité je dokončit.

North & South

■ Infogrames 1989

TEST 2 / 90		
Grafika	83 %	
Zvuk	81 %	Celkem
Idea	86 %	86 %
Přitažlivost	91 %	

NORTH & SOUTH je strategickou hrou z období občanské války v USA. Proti sobě zde stojí armáda Konfederace (Jihu) a armáda Unie (Severu). Vaším úkolem je přivést jednu či druhou stranu k vítězství.

Hlavní menu:

V tomto (vlastně jediném) menu můžete provést základní nastavení hry a volbu některých dalších aspektů. Především musíte zvolit, zda budete hrát sám, spolu s přítelem, či zda má počítač válčit sám se sebou. Pokud budete hrát sám, zvolte, zda budete ovládat armádu Severu či Jihu.

Obtížnost:

Obtížnost hry lze nastavit tak, že do čela obou bloků postavíte velitele na různé úrovni. Kvalita velitele se projeví v celkové strategii vedení války, v průběhu bitvy či v intenzitě obrany pevností.

Typ hry:

Můžete zvolit, zda chcete hrát hru čistě strategickou, či zda si přejete zařadit i arkádové sekvence (bitva, pevnost, vlak).

Výchozí pozice:

Můžete volit mezi čtyřmi výchozími pozicemi-roky z období 1861-1864. Tedy od historického počátku konfliktu až do jeho ukončení. Vaše pozice ve zvoleném roce odpovídá skutečným historickým událostem.

Nyní můžete nastavit tři další aspekty, které přímo ovlivňují průběh hry:

1) *Zda mají do konfliktu zasahovat Indiáni a Mexičané.* Pokud je tato volba zapnuta, může se stát, že armáda, která se nachází na mexickém či indiánském území, může být zničena. Tato území jsou: Texas, Kansas, Iowa a Oklahoma.

2) *Bude-li bráno v úvahu počasí.* Při aktivaci tohoto aspektu se bude po mapě pohybovat bouřkový mrak. Pokud se některá jednotka nachází na území, kde právě řádí bouřka, nemůžete v tuto chvíli provést její přesun.

3) *Příjezd posil.* U území North Carolina může přistávat loď, která bude na americký kontinent dopravovat posily. Vylodí se vždy jednotka té strany, které území North Carolina náleží.

Pokud jste s nastavením hry spokojeni, odpalte GO.

Na mapě vidíte železniční trať. Pevnosti (železniční stanice) jsou na ní vyznačeny červeně. Vlak pro vás znamená peníze, za peníze můžete najímat další vojáky. Proto musíte obsadit alespoň dvě pevnosti tak, aby mezi nimi mohl jezdit vlak. Vaše příjmy při jedné cestě vlaku závisí na počtu obsazených území.

Hra obsahuje tři arkádové sekvence:

Bitva:

Pomocí klávesy SHIFT přepínáte mezi skupinami vojáků, které budete ovládat (pěšáci, jízda, dělostřelci). Vaše jednotky se mohou spojovat. Jedna jednotka však může obsahovat maximálně 3 děla, 18 pěšáků a 9 jezdců. Tři děla můžete využít rovnou, pěšáků však může být na obrazovce maximálně šest, jezdci mohou být maximálně tři. Proto další vojáci vstupují do bitvy až po zničení prvních útvarů.

Pevnost:

Pokud útočíte, musíte se co nejrychleji probít na druhou stranu pevnosti, kde vztyčíte vlajku svého uskupení. Vyhýbejte se krabicím s dynamitem, psům a nepřátelským vojákům (obránce můžete likvidovat pěstí nebo nožem). Váš pokus o obsazení pevnosti je však časově omezen - spěchejte, jinak neuspějete! Pokud pevnost bráníte, musíte naopak zabránit nepříteli v cestě ke stožáru. Pohybem joysticku nahoru či dolů můžete umístit obránce na střechu či na zem. Zadržte protivníka za každou cenu!

Vlak:

Pokud vaše armáda obsadí území, přes které vede protivníkova železnice, můžete se pokusit o přepadení vlaku. Vaším úkolem je dostat se až k lokomotivě, vlak zastavit a vybrat z něj peníze. V případě útoku na váš vlak se ocitnete v roli obránce. Přepadávání vlaků je dobrým důvodem k použití l[illegible]é železniční tratě. Na dlouhé trase může nepřítel ohrozit vlak ze tří území, kdežto u krátké trasy se nepřítel může o přepad pokusit z jediného území, které lze pohodlně ubránit.

Strategie:

Strategie u této hry hraje zajisté velikou roli, byť není nijak zvlášť komplikovaná. Pokud však před začátkem hry zvolíte volbu příjezdu posil, bude výsledek hry záležet především na tom, která armáda obsadí území North Carolina. Tím je strategie hry poněkud zmechanizována. ■

TAD

NORTH & SOUTH je excelentně provedená jednoduchá strategická hra. Hudba i grafika jsou perfektní, zvukové efekty výborně propracované. Všechny arkádové sekvence se vyznačují velikou nápaditostí. Na této hře je příjemné především to, že se autoři zcela vyhnuli schématům a šli vlastní cestou.