



602

SPECTRUM

4

88



Náš seriál...

Grafické programy na ZX-Spectrum

=====

Abychom si od začátku rozuměli - dnes budeme mluvit pouze o těch grafických programech, jejichž výsledkem je plošný obrázek, například takový, jaký najdete na začátku většiny počítačových her.

Po svém uvedení na trh znamenalo Spectrum ve své cenové třídě převrat v počítačové grafice. Okouzlení uživatelé počítačů ZX-81 začali nesměle osahávat možnosti patnáctibarevné bodové grafiky a ta také byla hlavním apelem výrobce na zákazníky - v té době byla běžnější Spectra s 16K pamětí a takovou pamět' si mohl každý "jednaosmdesátkář" snadno připojit. Proto už na úvodní kazetě k novému počítači byl prajednoduchý program, umožňující tvorbu obrázků (Draw - 82 Psion). S tím se ale opravdu moc nakreslit nedalo. Mezeru na trhu vyplnil hned v následujícím roce program Melbourne Draw (83 Melbourne House). Jeho ovládání bylo těžkopádné, některé věci se s ním prostě udělat nedaly, ale přinesl dvě myšlenky, které se udržely ve všech novějších programech: možnost zvětšení části obrázku a dva informační řádky, které lze podle potřeby posunout tak, aby šlo kreslit po celé obrazovce. Je pozoruhodné, kolik pěkných titulních obrázků k hrám vzniklo s pomocí tohoto neohrabaného nástroje.

Značný ohlas na trhu vzbudil program Paintbox (84 Print'n'plotter). Ten dokázal pracovat s několika sadami uživatelem definované grafiky (opět myšlenka, která už nezapadla) a hlavně se prodával s připojeným velice působivým demonstračním programem. Jeho ovládání bychom označili dnes jako nepohodlné, počet funkcí začal narůstat do stádia, kdy už jejich vyvolávání nebylo možné bez stálého listování manuálem. Program byl sice později vylepšen (Paintbox 2), ale na této hlavní závadě už se nic nezměnilo. Předností byla možnost spolupráce s dalším programem téže firmy - Screen Machine (soubor strojových podprogramů pro práci s obrazovkou).

Ještě dále touto cestou se vydal program Leonardo (85 Creative Sparks). Ten už kreslil elipsy, dal se ovládat joystickem, zkrátka dokázal nakreslit téměř všechno. Ale za cenu naprosto nepřehledného ovládání, kdy uživatel neustále musel přeskakovat z jednoho módu do druhého, obraz se mu stále ztrácel pod pomocnými menu a v každém z mnoha módů měla většina tlačítek nějakou funkci. Málokdo tvořil počítačovou grafiku tak často, aby mu ovládání "přešlo do krve". Majitelům Specter v té době nezbyvalo, než závistivě pokukovat po počítačích pracujících podle zásad WIMP (Windows, Icons, Mouse, Pull-Down Menus - okna, ikony, myš, postupně rozváděná menu).

Prvním programem, který vycházel vstříc uživateli s alespoň naznačeným komfortem ovládání, byl Artist (85 Softechnics). Ten až na výjimky požadoval pouze volbu z menu v oněch dvou komunikačních řádcích (menu se tam střídala podle potřeb dvě). Pamatuji se, že zejména v našich podmínkách, kde manuál k programu je často vzácností, bylo jednoduché ovládání Artista rozhodující předností.

Téhož roku se však na trhu objevil program Art Studio (85 Rainbird/OCP). Ten už vycházel důsledně z principu WIMP, dokonce i s tou myší se prodával, lze ale připojit i joystick. Reklama výrobce halasila, že koupí programu se z majitele Spectra stane

majitel počítače Apple Macintosh. Skutečně - na první pohled je vidět, že tvůrci se inspirovali programem MacPaint (včetně vzorků pro plnění obrazců). Důslednou prací s okénky se podařilo odbourat řadu zbytečných funkcí - tak např. elipsu nakreslíte jako kruh, definujete kolem něj okénko a tomu pozměníte rozměry stran. Všem bylo jasné, že tento program ovládaný tak, že uživatel prakticky nemusí spustit zrak z obrazovky, odsunuje všechny ostatní do starého železa. Bylo to bohužel jasné i výrobci, který opatřil program všemožnými zajištěními proti kopírování včetně Lenslocku drásajícího nervy i poctivým zákazníkům. Samozřejmě netrvalo dlouho a objevily se kopírovatelné verze vytvořené našimi, maďarskými a jugoslávskými piráty. Obvykle z nich zmizelo ovládání myši nebo joystickem a stává se, že po delším provozu prostě vypadnou - to je zvláště veselé po dvouhodinovém kreslení nádherného obrázku. Je třeba přiznat, že těmito neduhy netrpí "pocestena" verze rozšiřovaná 666. ZO Svazarmu. Majitelům ostatních verzí lze jen doporučit, aby své výtvořky nahrávali průběžně na pásek tak, aby v případě havárie nemuseli začínat znovu.

Snahou přinést přece jen něco nového byli vedeni autoři programu Animator 1 (86 Softcat Micros). Lze s ním vytvářet "pohyblivé" obrazovky, tedy takové, kde se část obrázku pohybuje. Ovládání je pomocí menu, hlavní důraz je ale kladen na tvorbu "filmových okének" pro pohyblivé části obrazu. Nezdá se, že by to mezi majiteli Specter našlo nějakou zvláštní odezvu, už proto, že to uměl i Artist.

Pokud vím, je prozatím nejdokonalejším programem tohoto druhu Artist II (86 Softechnics/Edge). Nejedná se vlastně o program, ale o soubor programů. První z nich je samotný Artist II, který představuje podstatné vylepšení proti Art Studiu. Menu jsou doplněna dvěma řádky ikon, ovládání je z klávesnice, joystickem nebo dvěma druhy myši. Omezení tu už nejsou téměř žádná. Například (oproti Art Studiu) můžete "vystřihnout" jakýkoli, i nepravidelný tvar z obrázku a "přilepit" ho do jiné obrazovky. Stín ohraničíte "neviditelným" perem, takže po vyplnění vzorkem bude mít nerovné okraje atd. Vzorně je vyřešena spolupráce s perifériemi včetně disket a tiskáren. Bohužel mnoho funkcí je těžko zjistitelných metodou pokusů a omylů, takže přečtení manuálu je podmínkou pro toho, kdo chce program plně využít. Manuál se pokusíme uveřejnit na pokračování v tomto zpravodaji. K programu je připojen pohodlný Sprite Designer, s jehož pomocí docílíme přinejmenším takové výsledky, jako s výše zmíněným Animátorem 1. V seriálu jsme již hovořili o Pagemakeru, který kombinuje obrázky s textem z textového editoru The Writer. Není to žádný Fleet Street Editor, ale pro uživatele Spectra je to jediný program pro tzv. Desktop Publishing - výrobu ilustrovaných publikací "na psacím stole". Konečně je tu také výkonný Screen Compressor, který dokáže např. komprimovat šest digitalizovaných fotografií ze hry Samantha Fox Strip Poker do necelých 15K paměti (a zachovat přitom jejich rychlé a pohodlné vyvolání z BASICu).

Jestliže jsem v předchozích dílech seriálu radil, aby si každý zvykl na "svůj" program a nenechal se zviklat nepodstatnými výhodami třeba nového textového editoru, u programů pro kreslení to rozhodně neplatí. Vývoj je tu rok od roku velice patrný a práce s Art Studiem je mimo jakékoli možnosti srovnání například s Melbourne Draw. Chcete-li Vy (nebo Vaše děti) na Spectru kreslit, opatřte si to nejlepší, co můžete.

Počítačové hry jako společenský fenomén

=====

Jako snad každý z majitelů osobního počítače jsem od začátku měl a používal počítačové hry. Skutečně od začátku, protože iniciativní výrobce k mému Spectru přibalil osm kazet her. Sice to byly hry, nad kterými bych se dnes ošklíbal, ale znamenaly vlastně moje první setkání s profesionálně provedeným softwarem. Záhy jsem pronikl do světa assembleru a vyšších programovacích jazyků, textových editorů, databank, hudebních, grafických, matematických, statistických, výukových a jiných "vážných" programů. Mám je, používám je a nějaké jsem i spáchal. Víím však, že jen málo z nich (o těch mých už vůbec nemluvě) je dotaženo do takového stupně dokonalosti, líbivosti a toho, čemu se říká "user-friendliness", jako počítačové hry. Opravdu si myslím, že začátečník by se měl naučit základy programování v assembleru z nějaké dobré pomůcky, ale s těmito základy by se měl vrhnout do jednoduchých her a rozebrat je na kousky. Takovou neobyčejně vhodnou hrou je např. staříčkový Manic Miner, napsaný patnáctiletým autorem.

Poznal jsem majitele počítačů, kteří povýšeně tvrdili, že počítač mají na to, aby jim čas šetřil a ne kradl. Bez výjimky se však ukázalo, že někde v dolním šuplíku takového zdánlivého suchara leží kazeta s hrami, nad kterými se jejich majitel dokáže nečekaně rozparádit. Protože však tento majitel nemá představu o trhu a jeho novinkách, jedná se obvykle o hry neobyčejně stupidní.

Není tajemstvím, že hry pronikly i do klimatizovaných sálů, kde jsou provozovány velké "mainframe" počítače a že nejednou je počítačový čas sdílen zároveň národohospodářskými výpočty a mariášem.

Na burzách a schůzkách majitelů osobních počítačů představují počítačové hry nejvýznamnější položku. Tam, kde se pořadatelé snaží hry a hráče omezit nebo vykázat, vede to pouze k vytvoření rychle rostoucích skupinek v předsíni, na schodech a podobně.

A najednou se dovídám, že počítačové hry jsou něčím, co by mělo nebyť, že soudný člověk používá svůj počítač zásadně k něčemu jinému, že hra = odstřelování potvor a tudíž = měknutí mozku atd. atd.

Pravda, pokud budeme mluvit o jednom typu her, kterému se obvykle říká "arcade", dojdeme asi k závěru, že na myšlení není náročný. Hrdina se někudy a něčím probíjí, střílí, kope, škrtí a poskakuje. Ale pozor! Je náročný na postřeh a soustředění, tedy na vlastnosti, které bychom tak rádi viděli u svých dětí. Navíc připočtíme získanou zručnost při ovládnutí klávesnice a ještě jednu důležitou věc: vyrůstá generace, která od malička konzumuje zábavu z televizní obrazovky, ať už z vysílání nebo videa. Obrazovka se stává synonymem zábavy a ta podvědomě spočívá v ležérním zcepenění a nasazení znuděného výrazu. Najednou však dítě může aktivně zasáhnout do toho, co se na obrazovce děje. Zažije prohry, ale i těžce vybojované výhry. Myslím, že to je nakonec důvod, proč mnoho rodičů intuitivně počítač dětem koupí.

Ale ono je mnoho druhů počítačových her! Pravda, málokdy se vymyslí něco zbrusu nového. Je ale asi poněkud pošetilé smést třeba šachy do jednoho pytle s arkádami a označit nálepkou "hloupost". Přesně to však mnoho autorů dělá! Co třeba

simulátory, které mi umožní prožít let stíhačkou, vrtulníkem, jízdu formulí 1 či parní lokomotivou, plavbu ponorkou nebo rychlým torpedovým člunem, zkrátka věci, které bych bez počítače nezažil. Nebo strategické hry, ať už jde o válečné simulace, nebo hry typu "land management". A co teprve tzv. "adventures", textové hry, kde se ze mě stává spoluautor dosud nenapsané knihy o napínavém obsahu (a navíc se přitom řádně procvičím v angličtině).

Odmítám uznat tyto obvykle dokonale provedené ukázky programátorské práce za něco ostudného a nežádoucího. Počítačové hry jsou tu s námi, asi ještě dlouho budou a mají své opodstatnění. Snažme se tedy, aby majitelé počítačů měli přístup k tomu nejlepšímu a aby věděli co s tím. Právě u těch složitějších programů nevystačíme se zjištěním co je "vlevo", "vpravo" a "střílení". Proto uveřejňujeme návody ke hrám, proto se snažíme držet krok s novinkami a zpřístupnit je členům klubu. Uvítáme v tom i Vaši spolupráci.

Chyby v BASICu?

=====

BASIC vtrhnul do sdělovacích prostředků, do novin, časopisů, knih, magazínů. Zastihl nepřipravené redaktory, sazeče i korektory, kteří nechápou, že stačí jediná chyba ve výpisu programu a program prostě nebude fungovat. On nejenže nebude fungovat, on ještě potrápí čtenáře, který bude hledat chybu především u sebe ve špatném přepsání do počítače. V zahraničních časopisech proto zdomácněly programy, které automaticky pomocí kontrolních součtů kontrolují každý řádek programu. Dlužno přiznat, že prvenství asi patří počítačům Commodore. Nejvíce se nám líbí program Weryfikator II, otištěný v časopisu Bajtek s upozorněním, že redakce Bajtku bude přijímat k otištění pouze programy, opatřené kontrolními součty řádků. Náš zpravodaj vzhledem ke svému rozsahu upustil od publikace delších výpisů (programy šíříme nahráváním na schůzkách), přimlouváme se však za masové zavedení této praxe i u nás. Proto polský program autora Piotra Strzelczyka otiskujeme:

```
100 CLEAR 65268: LET nrlin=256:LET adres=65269
105 READ a$:LET l=LEN a$:IF l=0 THEN PRINT "O.K.":STOP
110 LET sum=0
115 FOR i=1 TO l-4 STEP 2
120 LET cislo=(CODE a$(i)-48-7*(a$(i)>""))*16+(CODE a$(i+1)-
    48-7*(a$(i+1)>""))
125 POKE adres,cislo:LET sum=sum+cislo:LET adres=adres+1
130 NEXT i
135 LET sump=(CODE a$(l-2)-48-7*(a$(l-2)>""))*256+(CODE a$(l-
    1)-48-7*(a$(l-1)>""))*16+(CODE a$(l)-48-7*(a$(l)>""))
140 IF sum <> sump THEN PRINT "chyba v radku c.":nrlin: BEEP
    1,0:STOP
150 LET nrlin=nrlin+1:GO TO 105
256 DATA "3E02CD01162A535CE5CD55183E20D73E58F"
257 DATA "20D73E14D73E01D7DDE1E5DD5E02DD56849"
258 DATA "03131313130600210000DD4E0009DD232AA"
259 DATA "1B7AB320F52B2B2B2B2B2B2B2B2B2B2B2BE5D1596"
260 DATA "7BCB3FCB3FCB3FCB3FC630FE3A3802C67D1"
261 DATA "07D77BE60FC630FE3A3802C607D73E146AC"
262 DATA "D73E00D73E0DD7E1DD214B5C7DDDBE007AC"
263 DATA "20967CDDBE012090C9DDE5F5C5D5E597D"
264 DATA "2A495CCD6E19204C23235E23562B2B2B42D"
265 DATA "13131313E5DDE1210000DD4E0009DD23544"
266 DATA "1B7AB320F52B2B2B2B2B2B2B2B2B2B2B2BE5D1596"
```



```
267 DATA "2E1C7BCB3FCB3FCB3FCB3FC630FE3A38753"  
268 DATA "02C607CDD0FF7BE60FC630FE3A3802C6809"  
269 DATA "07CDD0FFE1D1C1F1DDE1FFC9D5265836B16"  
270 DATA "4726405F1600EB292929ED4B365C09EB546"  
271 DATA "06081A77241C10FAD12CC9F3ED46FBC9799"  
272 DATA "C36EFFF33E3BED47ED5EFB1872E"  
513 DATA ""
```

Po úspěšném dokončení běhu programu můžeme dát příkaz NEW.

Program napsaný v BASICu opatříme kontrolními součty řádků tak, že použijeme příkaz RANDOMIZE USR 65269. Na adrese 65270 je číslo kanálu pro přenos takto upravených dat: POKE 65270,2 směruje výtisk na obrazovku, zatímco POKE 65270,3 na tiskárnu.

Druhá část programu využívá maskovaného přerušení a tiskne v pravém horním rohu obrazovky kontrolní součet BASICového řádku, ve kterém se nachází editační kurzor. Tuto funkci vyvoláme příkazem RANDOMIZE USR 65519.

Celý program je umístěn mezi adresami 65269 až 65535 a je nerelokovatelný.

Co s programem...

A R T I S T I I *****

1. Začínáme =====

Nahrání programu z pásku -----

Artist II sestává vlastně ze čtyř samostatných programů. Je to Artist II samotný, pak "Sprite & Font Designer", "Page Maker" a "Screen Compressor". Všechny tyto programy nahrajeme z pásku příkazem

LOAD "" (a stiskem tlačítka ENTER).

Program se pak nahraje a automaticky rozběhne. Před tím, než se rozběhne Artist II, zeptá se Vás, zda máte připojenou myš AMX. Máte-li, pak stiskněte "Y" a ENTER, jinak "N" a ENTER. Artist II nebude správně fungovat, zvolíte-li na začátku myš, která pak ve skutečnosti není připojena. Opačně, připojíte-li myš bez počáteční volby, nebude program fungovat. Proto hned na začátku pozor!

úvod -----

Jádrem programu Artist II je řada osvědčených grafických rutin, které proslavily jeho předchůdce, Artist I. To by ale nestačilo k tomu, aby časopis Sinclair User (listopad 1986) označil Artist II za "...vůbec nejlepší grafický program pro Spectrum". Artist II totiž ke snadnějšímu ovládní používá důsledně symboly (ICONS), okénka (WINDOWS), postupně vyvolávaná menu (PULL-DOWN MENUS) a řadu grafických rutin, se kterými se obvykle setkáváme až u počítačů cenové třídy nad 1000,-

Používání grafických symbolů, okének, menu atd. se vžilo u počítačů jako Apple Macintosh, Atari ST a později také u počítačů kompatibilních s IBM PC za použití pomocných programů

GEM, WINDOWS apod.

Počítače používající těchto prvků se staly velmi populárními, protože jejich obsluha je pro uživatele pohodlnější. Tyto počítače mají také obvykle připojenu "myš". Tu ovšem můžete připojit i k Vašemu Spectru. Víme zatím o dvou výrobcích - Kempston Mouse a AMX Mouse - oba budou fungovat s programem Artist II. Namísto řízení programu z klávesnice (například pohybu šipky po obrazovce pomocí kurzorových tlačítek) pak jednoduše pohybujete myší po hladké podložce vedle počítače v žádaném směru. Kurzor, nebo cokoliiv se má po obrazovce pohybovat, se pak pohybuje stejně jako myš po podložce.

Použití myši s programem Artist II naléhavě doporučujeme, protože snadnost ovládání programu se zvětší alespoň desetkrát.

Ale bez ohledu na to, jaký způsob ovládání zvolíte, je Artist II vymyšlen tak, aby Vám pomáhal při nalezení další správné volby. Po nahrání uvidíte prázdnou kreslicí plochu, horní řádek s menu a spodní řádek s grafickými symboly.

Váš dialog s programem asi nemůže být jednodušší, protože téměř ve všech případech se uskuteční pouhou volbou hesla z menu, nebo volbou příslušného grafického symbolu.

šipkový kurzor

Je základem používání programu. K vytváření grafik postačí, abyste věděli, jak ovládat pohyb kurzoru. Pak už stačí dvě tlačítka (nebo spoušť na joysticku, či knoflíky na myši), abyste plně kontrolovali běh programu.

Na obrazovce má šipka funkci ukazovátka. Při vlastním kreslení ukazuje, na jaké pozici vlastně kreslíme. Přesně také určí, odkud a kam budou nakresleny čáry, kružnice, čtverce atd. Rovněž umožní přesnou definici okének.

Chcete-li zvolit některou z mnoha možností, které program v daném okamžiku nabízí, stačí posunout kurzor na příslušné heslo v menu, nebo na grafický symbol a pak stisknout tlačítko. Jednodušeji to snad ani nejde.

Stručný přehled

Nejdříve musíme vědět, jakými tlačítky se program ovládá. Při použití myši (AMX nebo Kempston) nebo joysticku (Kempston) pohybujeme kurzorem prostě tak, že pohybujeme joystickem nebo myší. Na myši Kempston jsou dvě tlačítka, na AMX tři. Artist II potřebuje vždy pouze dvě tlačítka, takže pravé tlačítko na myši AMX není zapojeno. Tlačítko vpravo (na AMX prostřední) slouží ke kreslení a k výběru z menu. Levé tlačítko pak slouží ke zrušení volby a také k mazání pixelů na obrazovce. Rovněž ho použijeme k výběru některých dalších možností. Ovládání myši si procvičte. Budete překvapeni, jak to půjde rychle.

Na joysticku tlačítko spouště slouží ke kreslení, zatímco tlačítko "M" na klávesnici k mazání a výběru z menu.

Klávesnicí ovládáme program pomocí tlačítek:

- Q - nahoru
- S - dolu

I - doleva
O - doprava
M - kreslení a výběr z menu
N - mazání

Pohybujte kurzorem tak, aby byl v levém horním rohu obrazovky. Přitom zjistíte, že pohyb kurzoru je "inteligentní", jeho rychlost závisí na délce stisku tlačítka. Posuňte kurzor na heslo v menu, označené STORAGE (uchovávání).

Jak sám název napovídá, můžete v tomto menu zvolit některou z dalších položek, které umožňují nahrání nebo přehrání Vašich grafických výtvorů, jakož i jejich tisk. K tomuto menu se vrátíme později, prozatím si všimněte, že dva řádky nejsou černé ale zelené. Tyto položky v menu nefungují, pokud Artist II nezjistí, že je připojen Interface I nebo řadič disket Opus. Položky se týkají katalogizace diskety nebo cartridge a mazání souborů na nich.

K návratu z menu STORAGE postačí stisknout levý knoflík na myši. U klávesnice nebo joysticku jednoduše posunete kurzor stranou a uvidíte, že menu zmizí. Teď pohybem šipky vyvoláme menu TYPEFACE (druh písma).

Opět žádné překvapení. Podle očekávání můžete vybírat ze dvou velikostí písma a ze tří sad písma, uložených v paměti. Později si vysvětlíme, jak typ písma předefinovat.

Další heslo v menu je MODES (způsoby). Tady už budeme potřebovat více vysvětlování, i když s trochou znalosti angličtiny pochopíte, oč se asi jedná. V případě pochyb pak prostě vyzkoušejte, co se stane!

Podrobněji si o tomto menu řekneme dále, ale zatím postačí, že tady nastavujeme základní způsob, jak se bude kreslit. MODES musíte vyvolat vždy, když chcete změnit barvu, kreslit přerušovanou čáru místo plné, obrátit barvy inkoustu a papíru, provést s obrazovkou XOR a podobně.

Hned vedle je heslo SCREEN (obrazovka). Všechny položky z tohoto menu se vztahují k celé obrazovce, na rozdíl od okének. Když tedy například zvolíte CLEAR (mazání), vymažete celou obrazovku. VIEW (pohled) odhalí pohled na celou obrazovku bez menu a grafických symbolů.

Do hesla EXTRAS (zvláštnosti) si najedete kurzorem v případě, že chcete změnit způsob ovládání. Nevolte myš nebo joystick, nemáte-li tyto periferie připojeny. Odtud také můžete editovat pět sad písma i vzorky, používané pro malování ploch. Zkrátka, toto menu budete volit jen zřídka, ve zvláštních případech.

Poslední, ale jedno z nejdůležitějších, je heslo WINDOWS (okna). Schopnost Artistu II definovat na obrazovce okénka a pracovat s nimi, je jednou z jeho největších předností. Z tohoto menu určujete, co se uvnitř nadefinovaných okének stane a vztahy mezi okénkem a zbytkem obrazovky. Můžete měnit v okénku barvy, pohybovat v něm kresbou, pohybovat celým okénkem po obrazovce, otáčet okénko, nebo je zrcadlově převracet, zvětšovat ho i zmenšovat, spojovat připravenou kresbu s tou, která je na obrazovce.

Ještě se zběžně podívejme na grafické symboly v dolním řádku. Vlevo jsou symboly pro kreslení čar, kruhů, elips a

pravoúhelníků - obyčejných, nebo vyplněných. Vpravo pak je štětec, jehož velikost si můžete zvolit (nebo ho nahradit sprejem). Zvolený grafický symbol vždy změni svou barvu z bílé na černou. Dále vpravo jsou nůžky, pomocí nichž můžete oddělit nepravidelnou část své kresby a manipulovat s ní odděleně. Už jen tato možnost nadřazuje program Artist II nad ostatní grafické programy pro domácí počítače. Pak následuje lupa, s jejíž pomocí lze zvětšit část obrazu a provést jemné opravy. Vedle je kbelík s barvou, který použijeme při vybarvení ohraničené plochy barvou nebo vzorkem. Dál následuje symbol textu - písmeno "A". Text lze vkládat přímo do obrazu na jakémkoli místě. Tečkovaný čtverec slouží k definování okének.

Na závěr tohoto zběžného přehledu si všimněte pravého dolního rohu, kde jsou vidět zvolené barvy a vzorek pro vyplnění ploch.

Kreslení jednoduchého obrázku

Nejprve si zvolíme o něco tlustší štětec. Posuňte kurzor na spodní řádku a zvolte obrázek štětce. V horní části obrazovky se objeví menu, na kterém je zvolený štětec označen blikajícím modrým čtverečkem.

Vybereme si štětec o velikosti 3 x 3 pixely, to znamená ten, který je o dvě políčka vpravo od štětce nabízeného Artistem II po nahrání z kazety. Výběr provedeme myší, joystickem, nebo tlačítky I a O.

Kreslíme opět pohybem myši nebo joysticku a stisknutím spouště nebo pravého knoflíku. Z klávesnice pohybujeme štětcem pomocí tlačítek Q, S, I a O za současného stisku tlačítka M.

Mazat můžeme stiskem levého knoflíku na myši, nebo tlačítkem N.

Kreslení čar

Kreslení čar je velice snadné. Je-li obrazovka zaplněna výsledky Vašich předchozích pokusů, smažte ji volbou hesla SCREEN a položky CLEAR z menu. Pak zvolte horní ze dvou symbolů čar v levém dolním rohu obrazovky.

Po vymazání obrazovky uvidíte, že levý knoflík na myši (nebo 'N' na klávesnici) slouží k nastavení počátečního bodu linky. Teď můžete definovat čáru pomocí umístění křížku a následným stisknutím pravého knoflíku, nebo tlačítka 'M'. Tak můžete kreslit i čáry, které na sebe navazují. Jednoduše střídáte tlačítka 'N' a 'M', přičemž funkce kreslení čar zůstává navolena a nemusíte ji před každou čarou volit znovu. Ven se dostanete tak, že křížkem vyjedete do jiného grafického symbolu, nebo hesla v menu.

Dolní grafický symbol čar zvolte v případě, že chcete kreslit více čar, které začínají v jediném bodu.

(pokračování příště)

Něco pro jazykáře...

S C O P E =====

1. úvod -----

SCOPE je mnohaúčelový plně strukturovaný jazyk pro počítač SPECTRUM 48K. Pomocí 31 příkazů můžeme bez použití BASICu vytvořit program, který je kompilován nebo přepsán do částí paměti k tomu účelu rezervované.

Kompilovaný program je rozbíhán pomocí volání USR. Protože kompilovaný kód používá rutiny kompilátoru SCOPE, může pracovat pouze tehdy, je-li v horní části paměti program SCOPE přítomen. SCOPE je v první řadě směřován pro rychlé užití v grafice, barvě, zvuku a animaci.

Většina slov používá pouze celá čísla od 0 do 255. Pro body a tisk grafiky nepotřebujeme jiná čísla.

Vytváříte-li hru, potřebujete psát čísla od 0 do 65535 (např. skóre) a některá slova. Použití barev a zvuku je odlišné od BASICu. Proto si pečlivě prostudujte příslušné oddíly. Na následujících stranách je jazyk a jeho operace probrán krok za krokem. Tento manuál vychází z toho, že rozumíte svému počítači a máte prostudovaný manuál BASICu. SCOPE neobsahuje příkazy pro použití definované grafiky (UDG).

2. nahrávání -----

Zadejte LOAD "SCOPE".

Spustíte pásek a stisknete ENTER. Je-li nahrávání skončeno, je indikováno na obrazovce a po krátké přestávce je smazán obsah obrazovky a nižší část paměti, poté je připraven začít.

Nahrání programu SCOPE na pásek:

- a) nekompilovaný program - SAVE "jméno"
- b) kompilovaný program: použijte SAVE "jméno" CODE x,y. Začátek a délku bloku zjistíte dle popisu z odstavce 3.

3. uspořádání paměti -----

Než začnete psát programy v SCOPE, musíte vyhradit místo v paměti, kde bude uložen kompilovaný program. Zároveň musíte vyhradit místo, kde budou vloženy rutiny SCOPE. Po nahrání programu SCOPE je použitelná část paměti od 23755 do 59999 pro programování. Slova SCOPE zapisujte po Basicovém příkazu REM a proto musí SCOPE program začínat slovem ORG.

```
10 REM ORG; 40000, 50000
```

Předchozí řádek znamená, že kompilovaný program začíná od 40000 a rutiny SCOPE od 50000. Máte tedy volných 16 K pro BASIC (od 23755 do 39999), 10 K pro SCOPE (40000 až 49999) a 10 K pro rutiny SCOPE (50000 až 59999).

Tyto parametry jsou plně pod vaší kontrolou a kdykoliv je můžete změnit.

Občas se potřebujete dozvědět, jak velkou máte použitou část paměti. To se snadno dovíte, použijete-li následující pro BASICovou část paměti:

```
PRINT PEEK 23645 + 256 * PEEK 23646
```

a objeví se poslední adresa.

Pro SCOPE zadejte:

```
PRINT PEEK 60316 + 256 * PEEK 60317
```

Pro rutiny SCOPE zadejte:

```
PRINT PEEK 60318 + 256 * PEEK 60319
```

První slovo programu SCOPE musí být ORG a poslední EXIT.

4. syntaxe

Zde jsou uvedeny jednoduchá pravidla gramatické konstrukce slov SCOPE. Nedodržíte-li je, objeví se zpráva "Parameter Error".

1. Všechny příkazy musí začínat slovem BASICu REM.
2. Všechny příkazy se musí vypisovat. Nelze užívat žádné slovo z BASICu.
3. Všechny příkazy musí začínat velkým písmenem. Další písmena mohou být malá. Slova musí být napsána přesně tak, jak jsou ve slovníku.
4. Všechny příkazy musí být následovány bezprostředně středníkem
5. Operandy musí být odděleny čárkou.
6. Na řádku může být pouze jeden příkaz nebo slovo.
7. Stranou od slov, které nemají operandy, nebo následuje-li text, musí všechna slova končit dvojtečkou.

5. kompilace a rozbíhání programu SCOPE

Je-li řada příkazů programu SCOPE ukončena a má-li být kompilována do strojového kódu, musí být přepsána do části paměti vyhrazené programu SCOPE. Napište přímo tento příkaz:

```
PRINT USR 60450
```

a stiskněte ENTER.

Je-li v programu chyba, objeví se v levém horním rohu obrazovky zpráva. Není-li chyba, objeví se číslo poslední adresy SCOPE programu. Pomocí SCOPE můžete psát i příkazy v mnemonickém strojovém kódu.

Kompilovaný program rozbíháme pomocí příkazu RAND USR 40000 (byla-li tato část paměti zadána) nebo LET x = USR 40000. Nyní se podívejte do slovníku.

6. Slovník

ADD

Formát - Add; Bvar, číslo:

Vykonává - zvětšuje Bvar; uvedená hodnota

Příklad - 10 REM Add; A, 25:
 20 REM Add; c, 50:

Komentář - musí být použito pouze ke zvětšení Bvar, ne Var.

ATTR (atribut)

Formát - Attr; proměnná, číslo řádky, číslo sloupce
 nebo

 proměnná, proměnná
Vykonává - ukládá do proměnné určenné hodnoty barvy na souřadnici atributu

Příklad - 10 REM Attr; a, 10, 10:
 20 REM Attr; a, b, c,:

Komentář - číslo atributu barvy je číslo papíru a inkoustu dle tabulky barev.

BDR (border)

Formát - BDR; číslo nebo proměnná

Vykonává - mění barvu okraje. Hodnoty mohou být 0 až 7.

Příklad - 10 REM Bdr; 5:
 20 REM Bdr; a:

BVAR (big variable)

Formát - Bvar; písmeno, číslo

Provádí - upozorňuje počítač na změnu určité proměnné, bude obsahovat určenou hodnotu.

Příklad - 10 REM Bvar; a, 30000
 20 REM Bvar; C, 53000

Komentář - velká proměnná může být číslo od 0 do 65535. Může být až 52 velkých proměnných, úplně oddělené od Var. Písmeno A až Z nebo a až z může být použito pouze jednou.

CALL

Formát - CALL; písmeno

Provádí - instruuje počítač ke skoku na určitá následující návěští v místě rutin. Zajišťuje výstup z rutiny a návrat na následující příkaz.

Příklad - 10 REM Call; a:
 20 REM Call; B:

Komentář - musí být použito pouze pro návěští v místě rutin.

CHG (change)

Formát - CHG; číslo nebo proměnná

Provádí - mění pevný papír a inkoust a maže obrazovku

Příklad - 10 REM Chg; 16:
 20 REM Chg; a:

Komentář - čísla barvy papíru a inkoustu jsou kombinovaná - viz oddíl barev.

DEC (decrease)

Formát - DEC; proměnná, číslo

Provádí - zmenšuje určenou proměnnou

Příklad - 10 REM Dec; A, 5
 20 REM Dec; F, 10

DRAW

Formát - DRAW číslo barvy, + číslo, + číslo (nebo proměnná,

+ proměnná)
Provádí - kreslí přímku od posledního zadaného bodu ve specifické barvě, další číslo je počet bodů směrem doprava, je-li znaménko kladné, nebo doleva je-li záporné, další určuje plus nahoru a minus dolů.

Příklad - (př. proměnná Var; A = 50 a Var; B = 50)
10 REM Plot; 58, 50, 50:
20 REM Draw; 58, +A, -B:

Výsledkem je přímka od 50, 50 do 100,0.

END

Formát - END;

Provádí - sděluje počítači, že zde musí rutina končit a další řádky musí být napsané v programu SCOPE.

FSCR (fine scroll)

Formát - FSCR; číslo:

provádí - přikazuje scrollovat celou obrazovku směrem, který je určen číslem. Toto číslo odpovídá klávesám kurzoru.

5 - scroll vlevo
6 - scroll dolů
7 - scroll nahoru
8 - scroll vpravo

Komentář - kterékoliv jiné číslo je chyba.

EXIT

Formát - EXIT;

Provádí - ukončuje program SCOPE a vrací se do BASICu.

GET

Formát - GET; proměnná:

Provádí - ukládá kód poslední stisknuté klávesy do proměnné.

Příklad - 10 REM Get; k:
20 REM Get; K:

Komentář - Jestliže nebyla od poslední doby stisknuta žádná klávesa, žádný příkaz se nevykoná. Je to dobrá programátorská praktika použít proměnnou Var; k nebo K má stejný význam.

HALT

Formát - HALT; číslo nebo proměnná

Provádí - přikazuje zastavit program po dobu N televizních cyklů obrazovky. N může mít hodnotu 1 až 255, tedy je-li za 1 sekundu 50 cyklů, tak N = 255 je doba asi 5 sekund.
!! Nepoužívejte 0 - znamená to Halt navždy !!

Příklad - 10 REM Halt; 25:
20 REM Halt; 112:

INC (increase)

Formát - INC; proměnná, číslo

Provádí - zvětšuje danou proměnnou o zadané číslo

Příklad - 10 REM Inc; A, 5:
20 REM Inc; e, 12:

JUMP

Formát - JUMP; písmeno

Provádí - přikazuje počítači skočit na instrukci následující

dané návěští v místě programu.
Příklad - 10 REM Jump; A:
 10 REM Jump; b;
Komentář - Nesmí být použito návěští použité v rutině.

LABEL (návěští)

Formát - LABEL; písmeno:
Provádí - upozorňuje na další adresu, na kterou se skače v
 v programu. Musí předcházet příkazu skoku.
Příklad - 10 REM Label; B:
 15 REM Label; a:
Komentář - K použití 52 písmen, šá malých a 26 velkých. Návěští
 může být použito buď v místě programu a nebo v místě
 rutin. Samozřejmě nemůže být žádné návěští použito
 více než 1x.

LIM (limit)

Formát - Lim; Bvar, číslo, návěští
Provádí - umožňuje skok na návěští, je-li číslo rovno proměnné.
Příklad - 10 REM Lim; a, 0, B:
 10 REM Lim; D, 30, A:

MINUS

Formát - MINUS; Bvar, číslo
Provádí - zmenšuje Bvar
Příklad - 10 REM Minus; F, 50:
 20 REM Minus; c, 120:
Komentář - musí být použito pouze pro Bvar, ne Var

NOTE

Formát - NOTE; text:
Provádí - Stejně jako REM v BASICu. Umožňuje do programu vkládat
 komentář.
Příklad - 10 REM Note; to je čtverec:

NUM (number)

Formát - NUM; číslo barvy, Bvar, číslo řádky, číslo sloupce
Provádí - tiskne Bvar v barvě a na místě specifikovaném příkazem
Příklad - 10 REM Num; 56, z, 10, 10:
 20 REM Num; 40, E, 5, 8:

ORG (organize)

Formát - ORG; číslo, číslo:
Provádí - určuje počítači adresované pole pro program SCOPE a
 pole rutin.
Příklad - 10 REM Org; 40000, 50000:

OVER

Formát - OVER; 1 nebo OVER; 0
Provádí - 1 kreslí nebo píše přes původní obraz
 - 2 kreslí nebo píše, maže původní obraz
Komentář - Je-li stejný charakter vytisknut dvakrát, má to
 stejný účinek jako prázdná mezera.

PLOT

Formát - PLOT; číslo barvy, číslo řádku, číslo sloupce nebo

proměnná, proměnná, proměnná.

Provádí - Maluje bod v místě vymezeném souřadnicemi v dané barvě

Příklad - 10 REM Plot; 6, 100, 100:
20 REM Plot; c, 50, d:
30 REM Plot; a, b, c:

Komentář - podívejte se na detaily barvy - čísla.

PUT

Formát - PUT; číslo barvy, číslo řádky, číslo sloupce, text nebo
 číslo barvy, proměnná, proměnná, text

Provádí - ukládá na obrazovku zadaný text v dané barvě a na
 daném místě

Příklad - 10 REM Put; 5, 12, 10, text:

RND (random)

Formát - RND; proměnná, číslo, číslo:

Provádí - ukládá do dané proměnné náhodné číslo, přičemž platí
 následující pravidla: číslo je od druhého čísla včetně
 do součtu obou čísel.

Příklad - 10 REM Rnd; A, 7, 56:
 přiřazuje A hodnotu od 56 do 63.

ROUTINE

Formát - ROUTINE

Provádí - Práví počítači, že následující řádky mají být zapsány
 do prostoru rutin do té doby, než bude napsán příkaz
 END.

(pokračování příště)

Pro začátečníky...

Základy programování ve strojovém kódu
(pokračování)

=====

Další skupina instrukcí, které pracují s jednotlivými bity
jsou instrukce AND, OR a XOR. První dvě z nich by měl znát každý
programátor v BASICu. AND a OR jsou skutečně příbuzné s
BASICovými příkazy AND a OR.

AND, OR a XOR provádějí porovnání dvou bajtů po
jednotlivých bitech. Jeden bajt musí být v registru A, druhý v
nějakém jiném registru, nebo v paměti na adrese uložené v HL,
nebo jako operand za instrukcí ve strojovém programu. Výsledek
tohoto porovnání je uložen v registru A, kde nahradí předchozí
hodnotu. Výsledek má vliv na nastavení příznaků znaménka, nuly a
parity. Tyto instrukce jsou proto velice podobné aritmetickým
instrukcím, ovšem jejich funkce je spíše "logická" než
aritmetická. Prohlédněte si pravdivostní tabulku, která
zobrazuje výsledky porovnání dvou bitů logickými operacemi:

AND	0 AND 0 = 0
---	0 AND 1 = 0
	1 AND 1 = 1
OR	0 OR 0 = 0
--	0 OR 1 = 1
	1 OR 1 = 1

XOR	0 XOR 0 = 0
---	0 XOR 1 = 1
	1 XOR 1 = 0

Tak například porovnáváme dva bity pomocí funkce AND. Jeden z nich má hodnotu 1, druhý 0. Výsledek je 0. Porovnání stejných bitů funkcemi OR i XOR povede k výsledku 1.

Jak to všechno funguje v praxi, to vidíte na následujících příkladech, které si pozorně prostudujte:

AND	OR	XOR
00110101 = 43	01001000 = 72	11100000 = 224
10100100 = 174	00101001 = 41	01010001 = 81
-----	-----	-----
00100100 = 36	01101001 = 105	10110001 = 177

Ted se pokuste vyřešit další tři příklady:

AND	OR	XOR
01010101 =	10100110 =	11001010 =
00110011 =	11001100 =	01101100 =
-----	-----	-----

Můžete se ale ptát, k čemu je to všechno vlastně dobré. Prozatím si stručně řekneme:

AND používáme hlavně k nastavení určitých bitů uvnitř bajtu. Dejme tomu, že chceme resetovat bity 0 až 3, tj. nastavit je na hodnotu 0. Můžeme samozřejmě čtyřikrát za sebou použít příkaz RESET, ale mnohem jednodušší je použít příkaz AND 11110000. Protože bity 0 až 3 v operandu jsou všechny nulové, budou bity 0 až 3 ve výsledku vždy nulové, jak si můžeme ověřit v pravdivostních tabulkách.

Podobně je to s funkcí OR. Ta nastaví blok bitů tak, že všechny mají hodnotu 1. Například OR 00001111 zajistí, že bity 0 až 3 budou mít ve výsledku hodnotu 1.

XOR porovnává bity tak, že jsou-li stejné, výsledek je 0, jsou-li rozdílné, výsledek je 1. Když použijete v BASICu příkaz PRINT OVER 1, používáte vlastně XOR. Pixel v barvě inkoustu je 1, pixel v barvě papíru 0, takže si to na obrazovce můžete jednoduše vyzkoušet.

Operační kódy logických instrukcí uvádí tabulka:

registr/bajt:	A	B	C	D	E	H	L	(HL)	n
AND:	A7	A0	A1	A2	A3	A4	A5	A6	E6
OR:	B7	B0	B1	B2	B3	B4	B5	B6	F6
XOR:	AF	A8	A9	AA	AB	AC	AD	AE	EE

V této tabulce znamená 'n' operand, umístěný ve strojovém programu bezprostředně za instrukcí AND, OR nebo XOR.

(pokračování příště)

Sinclair 602, technický zpravodaj pro mikroelektroniku a výpočetní techniku. Vydává 602. ZO Svazarmu pro potřeby vlastního aktívu. Odpovědný redaktor Michael Mánek. Adresa redakce: 602. ZO Svazarmu, Wintrova 8, Praha 6, 160 41. Telefon: 32-85-63. Povoleno UVTEI pod evidenčním číslem 87 006. Cena 6,- Kčs dle č. 1030/202/86. Náklad 800 výtisků. Praha, říjen 1988.