POUŽITÍ POČÍTAČOVÉ GRAFIKY PŘI VÝUCE ALGORITMIZACE

Ivana Kolingerová, Jana Krutišová, Václav Skala
Katedra informatiky a výpočetní techniky,
Západočeská univerzita, Americká 42, 306 14 Plzeň
email
kolinger@kron.zcu.cs, krutis@kron.zcu.cs, skala@kron.zcu.cs

1. Úvod

Algoritmizace a základy programování se v současné době vyučují prakticky na všech středních a vysokých školách. Většina nám známých publikací na toto téma se věnuje převážně výkladu jednotlivých rysů toho či onoho programovacího jazyka, aplikaci vývojových diagramů, strukturogramů apod. Je nicméně otázkou, jaký postup při výuce algoritmizace volit, aby student nebyl nucen se zabývat několika problémy najednou. Musí řešit otázku návrhu algoritmu, jeho formalizace a verifikace jeho správné činnosti. To vše v okamžiku, kdy ještě zdaleka nemá "zažitý" programovací jazyk a kdy se v mnoha případech zabývá spíše otázkou, jak "myšlenku" vyjádřit pomocí jazyka než podstatou řešeného problému.

Podobně jako v uměleckých oborech je nutné studovat metody "starých mistrů", je v programování zapotřebí se zabývat již existujícími algoritmy. Každý si zajisté položí otázku, jak by skutečné studium "starých mistrů" a vlastně celá výuka algoritmizace a programování měly vypadat.

Každý programátor, který se někdy pokoušel o analýzu algoritmu jiného autora, jistě potvrdí, že mezi jeho přečtením a pochopením je značný rozdíl. Abychom algoritmu plně porozuměli, většinou se neobejdeme bez ruční simulace jeho chodu. Tato simulace je poměrně pracnou záležitostí, proto by bylo vhodné ruční práci nahradit užitím výukového programu. Zde také vidíme pole působnosti pro aplikace počítačové grafiky.

Z tohoto důvodu byl proveden experiment, kdy činnost známých algoritmů byla realizována prostředky počítačové grafiky a jejich běh demonstrován pomocí animace. Jednotlivé

Sborník
příspěvků
12. sympózia
o algoritmech
ze Sborník Algoritmy '93
sympozium Stará Lesná - Vys. Tatry
26 - 29. 4. 1993

programy pro animaci algoritmů byly vytvořeny v rámci předmětu Základy počítačové grafiky na ZČU jako jedna ze dvou prací zadaných v rámci semestru. (Dohromady takto vzniklo 26 programů).

Na obr.1 je zachycen obvyklý tvar grafického výstupu. Právě prováděné části jsou vhodným způsobem zvýrazněny, viz část obrazovky s programem a vývojovým diagramem, aktuální činnost se předvádí pomocí animace, viz výměna prvků ve spodní části. Zároveň se vypisují aktuální hodnoty proměnných, což umožňuje studentovi sledovat postup provádění jednotlivých algoritmů. Takto koncipovaná animace je vlastně speciálním případem výukového programu.

2. Požadavky na výukové programy

Výukovým programem nehodláme suplovat výklad vyučujícího, protože se domníváme, že lidský faktor je ve výuce nenahraditelný. Využití programů předpokládáme pro procvičení již vyložené látky. Z tohoto předpokladu se také odvíjejí požadavky týkající se obsahu a formy výukových programů.

K výukovému programu by měla být k dispozici podrobná dokumentace shrnující jak procvičované téma, tak metodu řešení, a vysvětlující ovládání programu a význam všech na obrazovce se vyskytujících identifikátorů. Tato dokumentace by měla být dostupná i z programu, nejlépe na stisk určité klávesy, nebo alespoň ve formě úvodní informace. Z hlediska efektivního využití času je vhodné umožnit při opakovaném použití programu vypuštění tohoto úvodu. Žádoucí by také byla možnost nastavení různých úrovní obtížnosti a rychlosti běhu.

Důraz by měl být kladen na interaktivní práci. Uživatel by měl být veden k aktivní účasti, nikoliv pouze odsouzen k pasívnímu přihlížení. Např. formou dotazu jej nutit udržovat pozornost na dění na obrazovce. V dokonalejším případě by uživatelovy zásahy mohly ovlivňovat i průběh zobrazované činnosti v mezích mírných modifikací.

Před skončením programu neopomineme vyhodnotit výsledky dosažené při kontrolních dotazech.

Kromě interaktivní formy je vhodné mít možnost spustit program s předem zadanými hodnotami.

Program by měl být snadno ovladatelný, pokud možno pomocí obecně užívaných kláves (např. <CR>, <ESC> apod.) - svoji tvůrčí invenci napřeme jiným směrem než je vymýšlení nových, "neotřelých" kombinací kláves. Při každém požadavku na interaktivní vstup by měla být k dispozici přehledná a jednoduchá nabídka možných odpovědí, nejlépe formou menu. Pokud možno se vyhýbáme přímým alfanumerickým vstupům.

Vždy je nutno předpokládat, že uživatel bude zkoušet i jiné varianty, než jsou mu nabízeny, nelze tedy počítat s bezchybnou obsluhou. Je nutno předejít zhroucení programu při chybném ovládání, a tedy chyby ošetřovat přímo v programu; neošetřené chyby, které se "propracují" až na úroveň operačního systému a vyvolají adekvátní systémové hlášení, jsou nepřípustné. Je třeba dokázat zpracovat i opakovaně nesprávné odpovědi a blokovat určité klávesy. Při chybné reakci uživatele umožnit volbu návratu do posledního správného kroku nebo na začátek programu, dovolit opravu překlepů a změnu jen té části dat, která byla zadána chybně. Na každou platnou uživatelovu volbu je třeba adekvátně reagovat, dát najevo, že byla správně pochopena a že se realizuje.

Při návrhu dialogu se vyhýbáme počítačovému slangu a snažíme se o srozumitelné a krátké formulace.

Jako stálí uživatelé počítačové grafiky musíme potvrdit oprávněnost požadavků vyhýbat se při návrhu grafického řešení programu užívání většího množství barev, ostrým odstínům a příliš kontrastním barevným kombinacím unavujícím zrak. Tři vzájemně sladěné, dobře odlišitelné, ale ne přehnaně kontrastující barvy obvykle postačí. Také blikání částí obrazu je zajímavé pouze při jednorázovém užití programu, ale při častějším zacházení působí nepříjemně.

Podobná střídmost je žádoucí i pro zvukové efekty. Pokud si jako autoři programu tento typ poškozování životního prostředí nedokážeme odepřít, měli bychom alespoň umožnit vypnutí těchto efektů.

3. Programové vybavení pro výuku algoritmizace

Než přejdeme k podrobnějšímu popisu realizovaného programového vybavení, je třeba znovu zdůraznit, že je zamýšleno jako

podpůrný prostředek pro zvládnutí publikovaných algoritmů, nikoliv pro tvorbu nových. Jeho autory jsou studenti III. ročníku oboru Elektronické počítače ZČU v Plzni. Je napsáno převážně v jazyce C, menší část v jazyce Pascal.

Jedná se o následující adresáře s programy :

ARABRIM - převod čísel z arabské do římské soustavy
DAMA - společenská hra
DIAG - cyklická záměna diagonálních prvků matice
GENIUS - společenská hra
HANOJ - problém hanojských věží
HORNER - Hornerovo schéma
INTEGR - výpočet určitého integrálu pomocí lichoběžníkové metody a metodou Gaussových váhových koeficientů
INTERPOL - výpočet interpolačního polynomu v Newtonové tvaru
MAXPRVEK - stanovení vektorů řádkových maxim a vektoru sloupcových maxim v rámci jediného průchodu danou obdélníkovou maticí
NASPOLY - násobení dvou polynomů
NEVILL - Nevillův algoritmus pro výpočet hodnoty interpolačního polynomu pro dané x
NULMAT - nalezení největší nulové čtvercové submatice v matici
OBVMAT - součet hodnot prvků matice po obvodech všech soustředných čtverců a výpočet průměrných hodnot prvků matice v těchto čtvercích
POSUV - posun dat v matici : v lichých řádkách vpravo, v sudých vlevo, data, která "přetekla" z řádku, postupují do uvolněného místa následujícího řádku, data z posledního řádku pak do uvolněného místa v 1.řádku
PRUNIK - průnik uspořádané a neuspořádané množiny
PRUNIK1 - průnik dvou setříděných množin
PRUNIK2 - průnik dvou nesetříděných množin
PRVCIS - nalezení prvočísel
RIDMAT - převod řídké matice na tabulku
RIMARAB - převod čísel z římské do arabské soustavy
RUBIK - skládání Rubikovy kostky
SPLINE - interpolace spline - funkcí
SNEK - magická matice
TELESA - algoritmy pro vykreslování těles

VEKTOR - zrcadlové převrácení vektoru celých čísel kolem osy symetrie

ZAMENA - vzájemná výměna prvků hlavní a vedlejší diagonály matice

Ukázka grafického výstupu na obr.1 pochází z programu VEKTOR. Je na ní zachycen okamžik, kdy dochází k výměně dvou prvků ve vektoru. Aktuální místo je ve výpisu programu i vývojovém diagramu od ostatních částí barevně odlišeno. Program je možné krokovat nebo nechat proběhnout celý. Jiné aktivní zásahy nejsou možné.

4. Shrnutí zkušeností z tvorby a distribuce výukových programů

Jak je vidět z předchozí ukázky, vytvořené programové vybavení splňuje zatím pouze část požadavků kladených na výukové programy. Většinou chybí podrobný návod, jednotné ovládání pomocí standardních kláves, možnost nastavení různých úrovní obtížnosti a především aktivní zapojení uživatele. V některých případech není dokonale ošetřen chybný vstup nebo potvrzena uživatelova volba.

Jsme si těchto nedostatků vědomi. Neradi bychom však značné množství práce, které by si další zdokonalení vytvořeného programového vybavení podle výše uvedených požadavků vyžádalo, vynakládali zbytečně, na dílo, které skončí v šuplíku, protože je nikdo nepotřebuje. Rádi bychom proto nejprve vyprovokovali odezvu té části veřejnosti, která se výukou algoritmizace zabývá, a může tedy nejspíše zhodnotit užitečnost celého projektu.

Ve jménu této snahy jsme poskytli 19 vybraným školám informaci o vytvořeném programovém vybavení. Bylo možné je získat zdarma po zaslání diskety. Jedinou podmínkou bylo krátké zhodnocení programů. Bohužel, hodnocení jsme získali pouze od nepatrného procenta všech respondentů. To jistě nedává příliš lichotivý obrázek zájmu zúčastněných pedagogů o nové formy práce.

Nechceme se však na podkladě negativních zkušeností se statisticky nevýznamnou částí naší pedagogické veřejnosti vzdát naděje na odezvu a spolupráci pedagogů. Jim především by měly

naše programy sloužit a na jejich reakcích a názorech tedy záleží, zda budeme na výukových programech dále pracovat anebo ne.

Možná, že cesta, po níž jsme se prvními krůčky vydali, nevede nikam; pak nám nezbyde, než konstatovat, že jsme zase skončili v cimrmanovské roli průkopníků slepých uliček, a zkusit kopat zas v některé jiné.

5. Závěr

Pokusili jsme se v tomto článku zamyslet nad problematikou užití počítačové grafiky při výuce algoritmizace a podělit se o první zkušenosti z tvorby a distribuce příslušného programového vybavení. Svůj přístup jsme dokumentovali na konkrétní ukázce jednoho výukového programu.

Prezentované programové vybavení bylo vytvořeno studenty III.ročníku FAV ZČU v Plzni jako podpůrný prostředek pro výuku algoritmizace. Zájemci mohou programy po zaslání diskety získat zdarma za podmínky, že svoje dojmy a zkušenosti z jeho užívání krátce zhodnotí podle doporučených kritérií a svůj posudek nám zašlou.

Zajímají nás především tato hlediska hodnocení jednotlivých programů :

- možnost využití v konkrétních učebních předmětech
- názornost
- způsob ovládání
- srozumitelnost doprovodných textů
- oblasti, kam by měl být zaměřen výběr algoritmů
- nedostatky

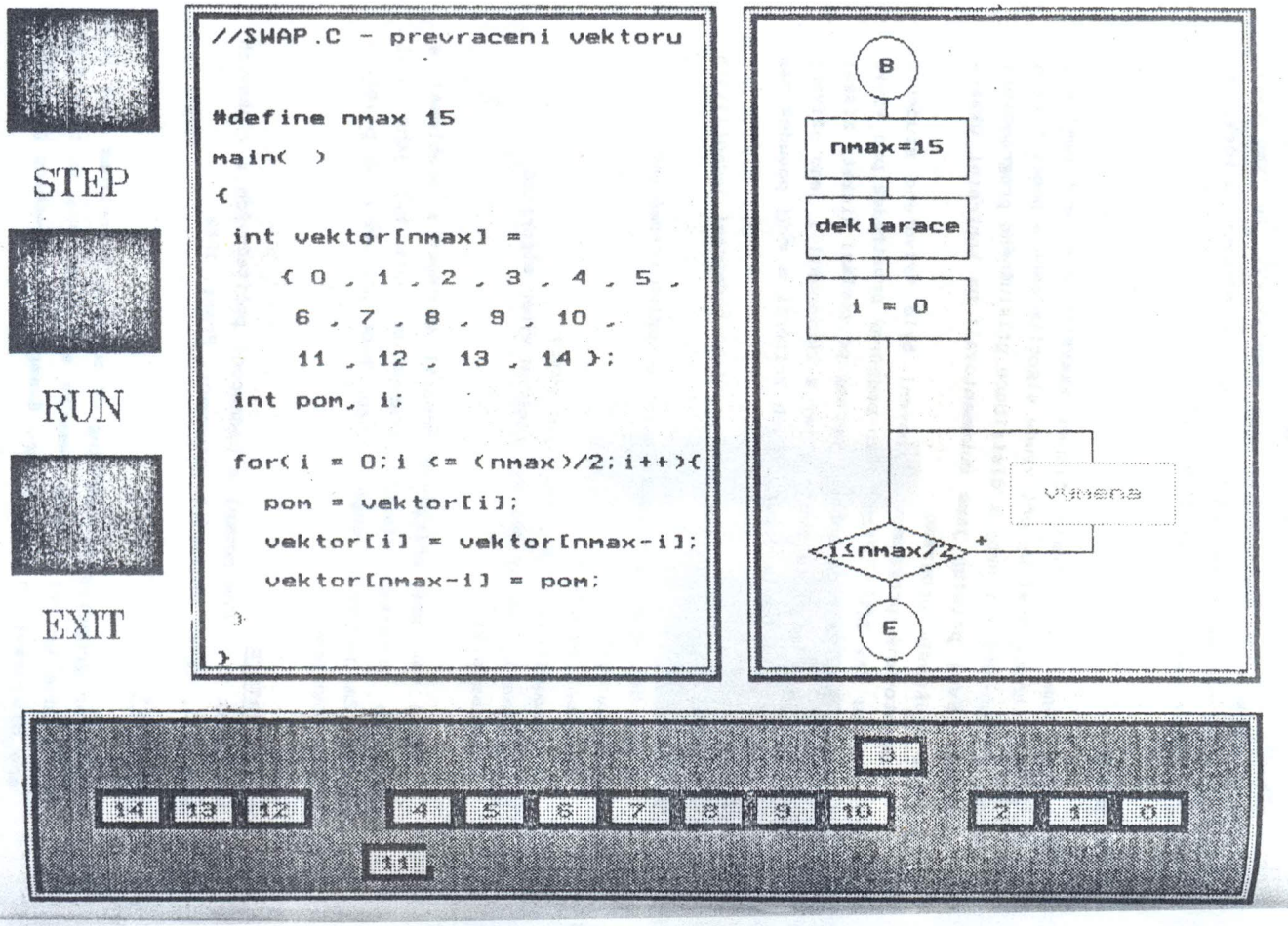
Pokud vás náš příspěvek donutil zaujmout k problematice výukových programů vlastní stanovisko, pak byl jeho účel splněn. Za jakýkoliv námět, názor nebo příspěvek do polemiky předem děkujeme.

6. Literatura

[1] Mazák E. : Posuzování a hodnocení počítačových a výukových programů, Ústav školských informací, Praha, 1989.

Abstract

Computer graphics in learning of computer algorithms - is it a good idea or not ? Some remarks and experience in this field are discussed in this paper. Example of a teaching program is included.



Obr.1 : Ukázka grafického výstupu programu VEKTOR