

Podle ohlasů některých uživatelů zatím není FreeCAD dostatečně stabilní pro nasazení u velkých zakázek, ale už jsem viděl jeho použití v profesionální praxi, které vypadalo velmi působivě. Každopádně je dostatečně kvalitní pro výuku práce s programy tohoto typu.

Program plánuji zařadit do výuky jako rozšíření deskriptivní geometrie.

FreeCAD je volně šířený pro systémy MS Windows, Mac OS X, Linux a další systémy typu UNIX. Uživatelské rozhraní komunikuje ve 24 jazycích včetně češtiny. Umožňuje číst a ukládat formáty FreeCAD (Festd), STEP, IGES, OBJ, STL, DXF, SVG, DAE, IFC, OFF, NASTRAN, VRML, DWG.

## Literatura

- [1] <http://povray.org/>
- [2] <http://www.wings3d.com/>
- [3] <http://www.blender.org/>
- [4] <http://www.makehuman.org/>
- [5] <http://freecadweb.org/>

# Násobík

*STANISLAV TRÁVNÍČEK*

Přírodovědecká fakulta UP, Olomouc

Skutečnost, že existují obrovské softwarové firmy, které vytvářejí obrovské množství dokonalého softwaru, by neměla být překážkou k tomu, aby si učitel matematiky nebo fyziky někdy vytvořil své vlastní počítačové programy, které mu mohou pomoci při přípravě na výuku nebo i při samotném vyučování. Výhodou takových programků je, že mohou být zcela konkrétně zaměřeny na potřeby autora, být relativně jednoduché a mít jednoduchou obsluhu. Například, když autor ví, že bude zadávat délky úseček, nemusí se při psaní programu starat o ochranu vstupu proti zadání písmene nebo záporného čísla, protože takový vstup jistě zadávat nebude.

V minulých letech jsem čtenářům MFI nabídl zdrojové texty několika takových programů z vlastní dílny. Například jako pomůcku při výpočtech lze použít [1] a [2], při grafickém znázorňování funkcí [3] a [4] a pro řešení slovních úloh s praktickými náměty [5], [6] a [7].

Jiná je situace, jestliže připravujeme program, který bude používat někdo jiný, tam už kontroly nesmí chybět, a ještě jiná, když připravujeme program, který mají používat žáci. Takové programy už musí být zcela odolné, protože žáci mohou s programem dělat cokoli. S jedním takovým programem z dílny poučeného amatéra se nyní seznámíme.

V posledních letech se vynořuje řada více nebo méně patřičně vzdělaných lidí, kteří aspirují na to, stát se novými Komenskými (dovolte mi tuto snad humornou větu). Většinou se brojí proti biflování (to zní přesvědčivě), ale tím kritici zčásti myslí to, že by se žáci vůbec neměli učit něco nazpaměť. Zdůrazňuje se přemýšlení, a že znalosti lze podstatně redukovat, protože přece máme internet, kde najdeme všechno, co potřebujeme. O tom všem by se mohla vést dlouhá debata, ale nám v této chvíli půjde jen o to, že něco žáci přece jen v hlavě mít musí. Já se domnívám, že k těmto prazákladům patří i malá násobilka, a to nejen odpovědi na otázky „kolik je  $5 \times 7?$ “, ale také „co je to 56?“ (a to nám kalkulačka ani internet neřeknou, i když například při sčítání zlomků se bez pohotové odpovědi na takové otázky neobejdeme).

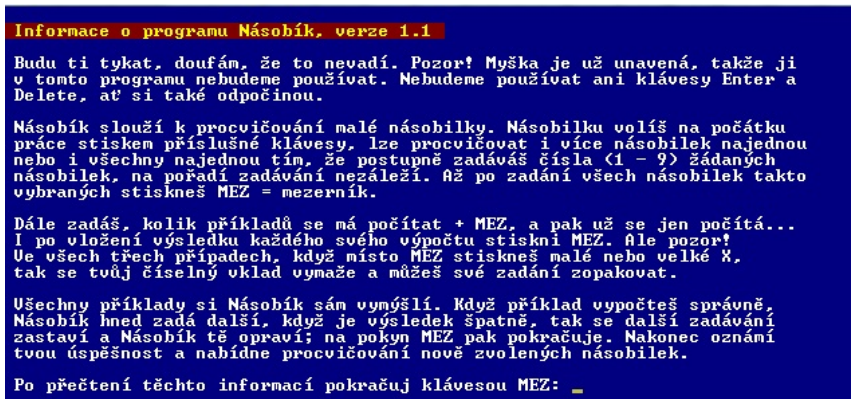
Sestavil jsem proto program *Násobilk*, který je určen pro procvičování (ne zkoušení) malé násobilky. Při takovém programu se musíme zabývat dvěma stránkami, jednak obsluhou při práci dětí s tímto programem a jednak strukturou programu, který má naplňovat nějaké didaktické záměry. V tomto pořadí nyní o *Násobilku* pojednáme.

Předně jsem si formuloval a co nejvíce uplatňoval ergonomické požadavky na program, se kterým mají pracovat i malé méně zkušené děti.

Program se rozběhne v okně, ne na celé obrazovce, které se dá přehlednou jediným pohledem. Tmavomodré pozadí šetří oči, bílým písmem jsou podávány potřebné informace, vlastní pracovní část má písmo žluté, takže umožňuje bezproblémové soustředění na práci; jen upozornění na chybu se správným výsledkem je výraznější (červená na bílé). Velikost písma je přiměřená, v případě problému u některého dítěte poradí školní informatik, jak písmo zvětšit. Pro maximální zjednodušení obsluhy jsou ze hry vyřazeny klávesy Enter, Del, Tab i všechny další (s výjimkou číslic, mezeríku a X nebo x), aby jejich používání nenarušovalo soustředění dítěte na vlastní práci. Jen na začátku a na konci práce se odpovídá písmenky

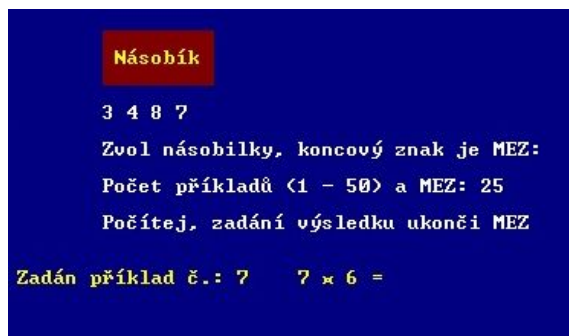
a či A nebo n či N na dotazy (ano–ne). Vzhledem k tomuto výraznému omezení funkcí kláves jistě každý uživatel programu přijde brzy na nabídnutý pohodlný postup, kdy levá ruka obsluhuje mezerník a pravá číslicovou klávesnici.

Po spuštění program nejprve na vyžádání sdělí uživateli všechny potřebné informace, (obr. 1).



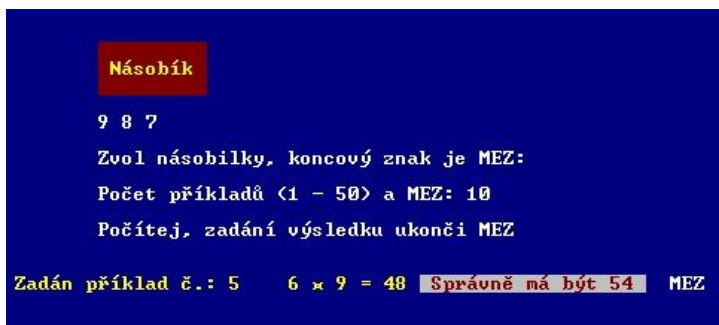
Obr. 1

Obr. 2 představuje obrazovku v průběhu práce. Vidíme, že je právě generován příklad z násobilky 7 a zcela úmyslně se přitom nedbá na pořadí čísel, aby toto pořadí nebylo důležité ani v myslech žáků.



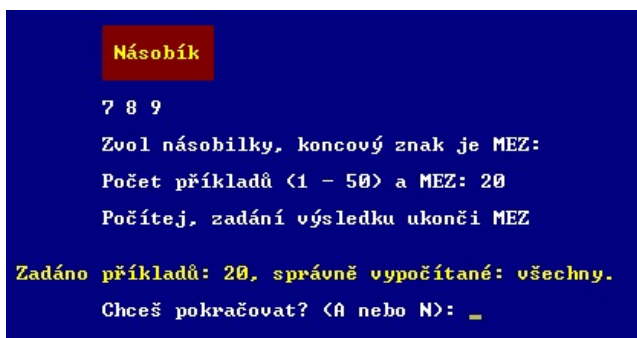
Obr. 2

Při chybném zadání výsledku Násobík na chybu upozorní (obr. 3).



Obr. 3

V průběhu procvičování násobilky zaznamenává počet správně vyřešených příkladů a na konci práce tento počet sdělí, na obr. 4 je sdělení v případě, že všechna řešení byla správně.



Obr. 4

Program byl vyzkoušen v několika operačních systémech a dostaly jej k ověřovacímu používání dvě Základní školy; dostaly i autorovo potvrzení oprávněnosti používání pro potřeby jejich žáků (tj. že nejde o „ukradený“ SW). Právě na základě tohoto ověření byla zavedena možnost, aby uživatel mohl, když se splete, ale včas to zjistí, svůj výsledek opravit (klávesou x nebo X).

Nyní o tom, jak pracuje nenovější verze 2.0. Činitele násobení tvoří program užitím generátoru náhodných čísel, ale vše se jen na náhodě nepochává, činnost programu je i mírně sofistikována; uveďme si některé prvky:

- Program pracuje i s nulou a desítkou, které však používá méně často než ostatní čísla 1 až 9; takové úlohy jsou snadné, ale vynechány být nemohly.
- Jak již bylo řečeno, úmyslně a nepravidelně zaměňuje pořadí činitelů, např. při násobilce sedmi může být sedmička prvním činitelem nebo jindy i druhým.
- Dále program nepřipouští, aby následující příklad byl přesně stejný jako předchozí.
- Pokud se zadavatel spletl například při násobení  $8 \times 7$  a odešle chybný výsledek, pak, kromě toho, že mu program sdělí správný výsledek, dostane zakrátko během další práce kontrolní zadání  $7 \times 8$ , jestli už správný výsledek zná.

Je samozřejmé, že program, který má zabezpečovat odolnost svého chodu, dá více práce a přemýšlení a má i delší zdrojový text; na publikování zde je příliš dlouhý a nakonec by čtenáři jiné poznatky, než zde byly uvedeny, nepřinesl.

Na začátku článku jsme si řekli, že poučený amatér si pro svou potřebu může tvořit i nedokonalé programy (vynechání kontrol), které mu dávají potřebné výsledky, a na příkladu programu Násobík jsme ukázali druhou krajnost – pokus o „dokonalý“ program z hlediska věcného i ergonomického.

## Literatura

- [1] *Trávníček, S.*: Vyčíslení aritmetického výrazu. MFI, 14 (2004/05), č. 6, s. 363–367.
- [2] *Trávníček, S.*: Číselné operace s velkými celými čísly. MFI, 16 (2006/07), č. 7, s. 424–436.
- [3] *Trávníček, S.*: Přímka daná dvěma body. MFI, 17 (2007/08), č. 7, str. 433–439.
- [4] *Trávníček, S.*: Lomené čáry jako grafy funkcí s absolutními hodnotami. MFI, 18 (2008/09), č. 7, s. 426–433, č. 8, s. 498–503, č. 9, s. 559–564.
- [5] *Trávníček, S.*: Slovní úlohy o celku a částech. MFI, 20 (2010/11), č. 3, s. 165–174.
- [6] *Trávníček, S.*: Úlohy o směsích. MFI, 20 (2010/11), č. 6, s. 361–369.
- [7] *Trávníček, S.*: Úlohy o společné práci. MFI, 21 (2011/12), č. 7, s. 428–438.