
Výukové programy a autorské prostředky

Jan Preclík, MFF UK Praha

Rozhodneme-li se používat počítač i při výuce "neinformatických" předmětů (biologie, zeměpis...), budeme k tomu potřebovat, mimo jiné¹, i vhodné **výukové aplikace** (anglicky *educational software* nebo *courseware*). Výukový program můžeme koupit už hotový (případně jeho tvorbu někomu zadat) nebo se pokusit takový program vytvořit sami.

V dnešní době je na našem trhu relativně dostatek různých výukových programů. Některé z nich ale nejsou příliš kvalitní nebo nemusí přesně vyhovovat našemu záměru – např. na určitou partii látky nekladou takový důraz jako my. Nevýhodou může být i jejich vysoká cena. Velké množství kvalitních zahraničních výukových aplikací není lokalizováno², tj. přeloženo do češtiny, a vzhledem k velikosti našeho trhu tomu tak pravděpodobně bude i nadále. Nechat si vytvořit program na zakázku je většinou zcela mimo finanční možnosti jednotlivce i školy.

Nevyhovují-li nám prodávané programy, můžeme se pokusit vytvořit program vlastní. Budeme k tomu potřebovat především nápad, odborné a pedagogické znalosti, spoustu času a nadšení, a také některý z programů (softwarových nástrojů), který nám tvorbu umožní a usnadní. Programy určené pro tvorbu výukových aplikací se obecně nazývají **autorské prostředky** (*authoring tools*).

Výukové programy

Ještě než se začneme zabývat autorskými prostředky, pokusíme se uvést stručný přehled různých typů výukových programů, abychom si udělali představu, o tvorbu jakých programů se můžeme pokusit.

Tento přehled nemá v žádném případě sloužit jako výčet všech na našem trhu dostupných programů, konkrétní programy jsou uváděny pouze pro ilustraci daného typu programů.

1. Výklad

Programy zaměřené na výklad (prezentaci nové látky) jsou převážně založeny na bázi **hypertextu**, často mají podobu WWW stránek. Výhodou hypertextu je, kromě možnosti jednoduše kombinovat text, grafiku, animace, zvuk..., i **nelineární prezentace učiva**. Informace nejsou studentovi předávány "lineárně" jedna za druhou jako v knížce, ale jednotlivá související témata jsou vzájemně propojena pomocí **odkazů** (*links, hyperlinks*). Takto strukturované učivo si student lépe a rychleji osvojí (neboť struktura hypertextového materiálu údajně napodobuje způsob, jakým jsou informace ukládány v lidském mozku) a

¹ Jedním z největších problémů je nedostatek počítačů na školách. Většina škol má (podle velikosti) jednu nebo dvě relativně dobře vybavené počítačové učebny, ve kterých se ale neustále učí informatika a výpočetní technika. Učitelé ostatních předmětů, i když by měli zájem, se k počítačům se svými studenty o svých hodinách zpravidla nedostanou.

² Lokalizovat program neznamená pouze přeložit veškeré texty do češtiny, ale také (a na to se často zapomíná) uzpůsobit chování programu národním zvyklostem (různé způsoby zápisu data a času, desetinný oddělovač ...).

snadněji si vytvoří potřebné vazby mezi souvisejícími tématy. Velkou výhodou hypertextových materiálů je při použití jazyka **HTML** možnost prezentace na Internetu a zpřístupnění materiálů velkému počtu zájemců z celého světa.

2. Zkoušení a testování

Pravděpodobně nejrozšířenějším typem výukových programů jsou zkoušecí a testové programy zaměřující se na ověření zvládnutí učiva. Takovéto programy by měly žákovi poskytovat **okamžitou zpětnou vazbu**, tj. prozradit, zda odpověď byla správná či nikoliv a v případě chyby prozradit správnou odpověď (nejlépe i s vysvětlením) tak, aby se žák při testování zároveň učil.

Nejjednodušší programy tohoto typu disponují pouze otázkami s **výběrovou odpovědí** (z nabízených možností vybrat jednu nebo všechny správné), složitější dovedou pracovat i s **odpovědí tvořenou** (kde žák nemá na výběr z několika možností, ale musí odpověď sám sestavit). Tvořené odpovědi se dosti obtížně vyhodnocují a způsob, jakým je jejich vyhodnocování v programu realizováno, často vypovídá o celkové kvalitě programu³. Dalším typem otázek jsou **otázky přiřazovací** (úkolem je k sobě přiřadit podle zadaného klíče pojmy z několika skupin nebo dané pojmy rozřadit do skupin; například přiřadit k sobě státy a jejich hlavní města), **uspořádací** (uspořádat dané pojmy podle určitého kritéria – seřadit historické události od nejstarší po nejmladší) nebo **umíst'ovací** (umístit pohoří na správná místa do slepé mapy).

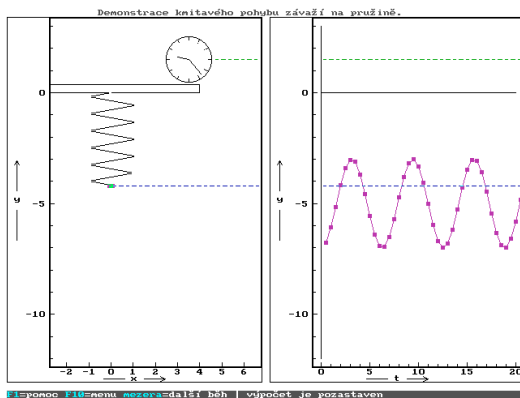
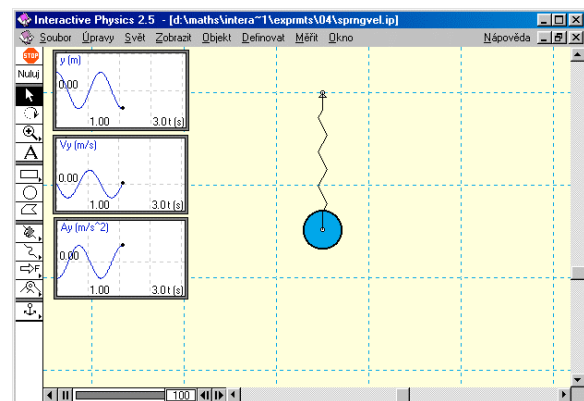
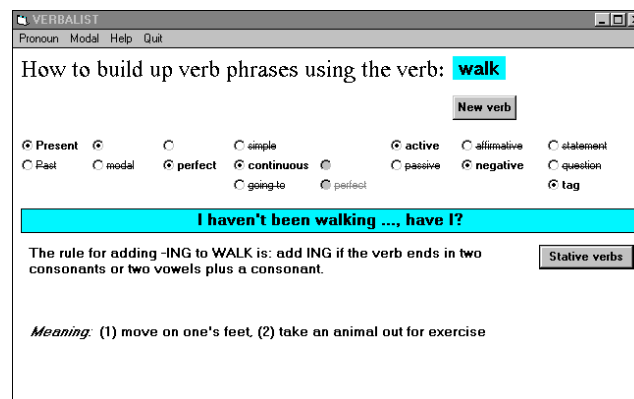
3. Simulace

Programy zaměřené na simulaci určité reálné situace mají použití především v přírodních vědách, nejčastěji v matematice (průběh funkce, trigonometrie) a ve fyzice. Žák může po změně parametrů ihned sledovat, jaké změny v simulovaném prostředí nastanou. Takto lze studentům přiblížit i situace, se kterými se v běžném životě nemohou setkat (např. řízení výkonu jaderného reaktoru).

Příklady takových systémů:

- ◆ Matematika a fyzika
 - **Famulus** (<http://vydra.karlov.mff.cuni.cz/bobo/fyzika/famulus/default.htm>), viz obr. 1
 - **Interactive Physics** (<http://www.krev.com/products/ip.html>), viz obr. 2
 - java applety na <http://www.physics.otago.ac.nz/Physics100/simulations/Gamelan>
- ◆ Angličtina
 - **Verbalist** – simulace stavby anglické věty, viz obr. 3
(<http://omicron.felk.cvut.cz/cgi-bin/charset/~bobr/bbaktvyb.html?charset=CP1250>)

³ Vyhodnocování správnosti tvořené odpovědi bývá nejčastěji realizováno na principu vyhledávání klíčových slov v textu odpovědi – určí se slova, která se ve správné odpovědi musí vyskytovat a případně i slova, která se tam vyskytovat nemohou nebo která budou při vyhodnocování ignorována (např. pomocná slovesa apod.). Díky bohatosti a tvárnosti českého jazyka však ani tento způsob nezaručuje bezproblémové a správné vyhodnocování.

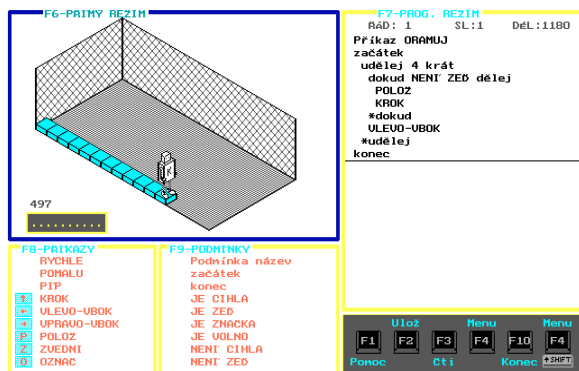
Obr. 1 – *Famulus*Obr. 2 – *Interactive Physics*Obr. 3 – *Verbalist*

4. "Dětské" programovací jazyky

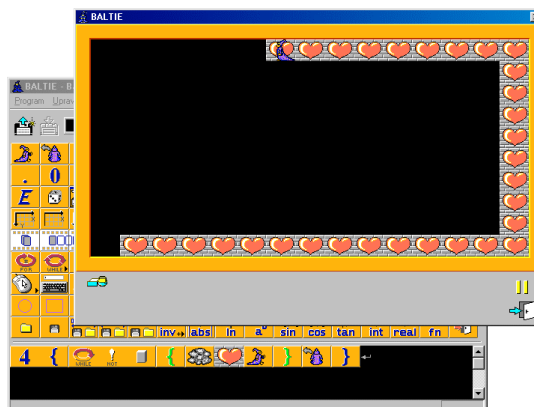
Pojmem "dětské" programovací jazyky označujeme prostředí sloužící k výuce programování a základních programátorských technik. Tato prostředí jsou určena pro naprosté začátečníky (děti předškolního a školního věku), proto je zde žádoucí přítomnost grafiky a příjemné ovládání, které by mělo žáky zaujmout. Většinou se zde žák nejprve učí ovládat (např. myši nebo kurzorovými klávesami) nějaký objekt (robota Karla, želvu v Logu, kouzelníka Baltíka) na obrazovce. Poté se přejde k možnostem programového ovládání, založeného např. na rekurzi – **Karel**, **Logo** (<http://www.eurologo.org>, <http://www.edi.fmph.uniba.sk/logo>, <http://center.fmph.uniba.sk/~kvi/slo/logo/comlogo.htm>) nebo strukturovaném programování – **Baltík**, **Baltazar** (<http://www.sgp.cz>).

Ukázky programů:

- ◆ Obr. 4, 5 – robot Karel obejde dokola celý dvorek a po obvodu položí značky
- ◆ Obr. 6, 7 – kouzelník Baltík obejde celou hrací plochu kolem dokola, po obvodu plochy vyčaruje srdíčka



Obr. 4 – 3D Karel



Obr. 6 – Baltík

```
Příkaz ORAMUJ
začátek
udělej 4 krát
dokud NENÍ ZEB dělej
POLOZ
KROK
*dokud
ULEVO-UBOK
*udělej
konec
```

Obr. 5 – ukázka
programu v jazyce
Karel



Obr. 7 – ukázka programu v systému **Baltík**

5. Výukové programy kombinující několik popsaných principů

Moderní výukové programy většinou kombinují několik výše uvedených principů, typicky spojují výklad s procvičováním a případným doporučením, které kapitoly je nutno zopakovat, nebo se simulací probíraných dějů. U rozsáhlejších systémů je vhodné, aby na začátku existoval *vstupní test (pretest)*, na jehož základě bude studentovi doporučen postup předkládanou látkou (např. s vynecháním těch pasáží, které už student ovládá). Příkladem takového systému je sada programů pro výuku angličtiny **LANGMaster** (<http://www.epaonline.com>), především kurz pro přípravu ke zkoušce *TOEFL*.