PROJECT SCRATCH

Michal MUSÍLEK

Abstract: The paper focuses on a children's programming language named SCRATCH and community project with the same name. It evaluates not only its unique strengths, but also indicates some reserves. The paper presents several examples which explain how to work with an integrated development environment and create the program composited from the prepared command blocks. This is similar to work with a quality building kit. The attention is aimed at work with graphic elements "sprites" and intuitive multi-threaded programming, in which individual scripts communicate with each other via messaging.

Key words: programming, algorithms, project, children's programming language, structogram, community site, social network.

PROJEKT SCRATCH

Resumé: Příspěvek představuje dětský programovací jazyk SCRATCH a stejnojmenný komunitní projekt, hodnotí jeho jednoznačná pozitiva, ale odhaluje i jisté rezervy. Na několika drobných příkladech ukazuje způsob práce s vývojovým prostředím, tvorbu programu jako sestavování strukturogramu z připravených příkazových kostek, které se podobá práci s kvalitní stavebnicí. Věnuje pozornost práci s grafickými prvky „sprity“ a naznačuje intuitivní vícevláknové programování, při němž jednotlivé skripty vzájemně komunikují prostřednictvím rozesílání zpráv.

Klíčová slova: programování, algoritmizace, projekt, dětský programovací jazyk, strukturogram, stavebnice, komunitní web, sociální síť.

1 Programovací jazyk SCRATCH

Programovací jazyk SCRATCH byl stejně jako jeho slavný předek, programovací jazyk LOGO, vyvinut na MIT (Massachusetts Institute of Technology), skupinou vedenou profesorem Mitechelem Resnickem [1]. SCRATCH vychází z jazyka LOGO, ale současně přináší celou řadu nových prvků:

- dokonalou intuitivnost ovládání vývojového prostředí a sestavování programů,
- lokalizace prostředí a příkazů jazyka do národních jazyků, včetně českého jazyka,
- samozřejmou, implicitní vícevláknovost,
- přirozenou objektovost,
- SMS spojení mezi objekty,
- ovládání událostmi
- a zejména nahrazení nutnosti znát příkazy jazyka a jejich přesnou syntaxi skládáním programů z připravených stavebnicových prvků (kostek).

Na druhou stranu je jazyk SCRATCH ve srovnání s jazykem LOGO nejen zjednodušen, ale v některých směrech také omezen. Hlavními omezeními jsou:

- nemožnost vytvářet své vlastní procedury a funkce,
- nemožnost používat rekurzi.

Obě výše uvedená hlavní omezení eliminuje projekt BYOB (Build Your Own Block) [2], který navazuje na SCRATCH a vyvíjí jej tým na Univeristy of California, Berkeley. Na rozdíl od SCRATCH však BYOB nemá lokalizace do národních jazyků, tj. v našem případě do českého jazyka. To je u jazyka, který by měl být vhodný pro výuku algoritmizace a programování již u žáků v mladším školním věku (7 až 12 let), dost podstatná nevýhoda. Proto se budeme nadále zabývat pouze jazykem SCRATCH.

Rozhodně stojí za zmínku, že SCRATCH převzal z jazyka LOGO kompletní želví grafiku, stejně jako práci s proměnnými a seznamy, takže zkušenosti s programováním v jazyce LOGO [3] se učitelům rozhodně neztratí.

Pro srovnání se podívejme na sedm výhod jazyka SCRATCH, jak je ve své diplomové práci [4] uvádí Alena Halousková, podle které jazyk SCRATCH:

- patří mezi vizuální programovací jazyky,
- je dostupný v české verzi,
- obsahuje dostatek programátorských konceptů pro první seznámení s programováním,
- je oblíbený u dětí (většinu tvůrců programů tvoří uživatelé ve věku od 10 do 20 let,
- umožňuje nahrávání projektů na internet,

- umožňuje snadné zapojení multimédií,
- je zdarma.

Kolektiv vývojářů prostředí SCRATCH v článku „Scratch: Programming for All“ [5] připomíná slova Seymoura Paperta, který řekl, že programovací jazyk by měl mít nízkou podlahu (být snadný pro začátečníka) a vysoký strop (po čase by měl umožnit také tvorbu náročnějších komplexních projektů). Měl by ovšem mít také široké zdi (podporovat mnoho rozdílných typů projektů, protože mají mnoho rozmanitých zájmů a stylů učení). Současné splnění těchto tří požadavků není vůbec jednoduché, ale zdá se, že jazyk SCRATCH je v tomto ohledu velmi zdařilým produktem.

2 Projekt SCRATCH

Jak naznačuje název článku, SCRATCH není jen programovací jazyk, ale komplexní projekt. Bonusem spojeným s programováním v jazyce SCRATCH je zejména specializovaný komunitní web a s ním spojená sociální síť. Komunitní web je stejně jako vývojové prostředí lokalizovaný, takže např. motto, či slogan komunity vesměs mladých amatérských programátorů *Imagine – Program – Share* je v české verzi přeloženo jako *vymysli – programuj – poděl se* [6]. Sociální síť umožňuje ukládání vlastních programů, stažení, prohlížení a úpravy programů kolegů, stejně jako k nim umisťovat záložky, komentovat, označovat je jako oblíbené, či dokonce milované programy. Programy (projekty) lze prohlížet přímo na webu, stačí mít webový prohlížeč s nainstalovanou podporou Java appletů. Vybrané kolegy si uživatel může označit jako přátele.

Projekt SCRATCH svým přístupem ke sdílení nápadů prostřednictvím komunitního webu podporuje myšlenku tvorby Open Source Software a tento postoj potvrzuje i tím, že zdrojový kód vývojového prostředí SCRATCH je také uvolněn jako Open Source na stránkách projektu.

Učitelé mají v rámci projektu své vlastní diskusní fórum, ve kterém se mohou podělit o zkušenosti s využitím vývojového prostředí ve výuce.

Pro českou komunitu dětských programátorů i pro jejich učitele připravila Alena Halousková českou on-line učebnici [7]

3 Programování jako stavebnice

Již v úvodu jsme se zmínili, že jednou z hlavních výhod ve SCRATCHI je sestavování programů z připravovaných kostek. Díky tomu nejen nemusíme znát tvar příkazů jazyka, počet a

význam jejich parametrů, ale současně s tvorbou programu vytváříme jeho vhodné grafické znázornění – strukturogram.

Ve strukturogramech SCRATCH hraje roli jednak barva (podobně jako v klasických českých strukturogramech, pojmenovaných *kopenogramy* podle svých tvůrců, kterými jsou Jiří Kofránek, Rudolf Pecinovský a Petr Novák), jednak tvar jednotlivých kostek.

Tvar kostky (viz obr. 1) určuje jejich základní funkci:

- hranaté kostky s výstupkem a obloukovým horním okrajem pro příkazy spouštějící jednotlivé skripty (příkazy typu START),
- hranaté kostky se zářezem a rovnou spodní hranou pro příkazy ukončující jednotlivé skripty (příkazy typu STOP) a nekonečný cyklus,
- hranaté kostky s výstupkem a zářezem pro výkonné příkazy a řídicí příkazy podmínek a konečných cyklů,
- oválné kostky (tj. pruhy s kulatým levým a pravým zakončením) pro funkce vracející číselnou hodnotu, proměnné a seznamy
- a šestiúhelníkové kostky (tj. pruhy se špičatým levým a pravým zakončením) pro funkce vracející logickou hodnotu (ANO/NE).

Obr 1: Skript pro ovládání pohybu sprite šipkami

Barva kostek popisuje, do které skupiny příkazů daný příkaz patří:

- tmavě modrá označuje skupinu **Pohyb**,
- lila (fialová) označuje skupinu **Vzhled**,
- sytě růžová označuje skupinu **Zvuk**,
- tmavě zelená označuje skupinu **Pero**,
- okrová (žlutá) označuje skupinu **Ovládání**,
- světle modrá označuje skupinu **Vnímání**,
- světle zelená označuje skupinu **Operátory**
- a oranžová označuje skupinu **Proměnné**.

Ukažme si to příkladem použití příkazových kostek na scénáři, ve kterém postava (*sprite*, *angl. skřítek, víla*; *implicitně však maskot programu – kocour*) okamžitě reaguje na povely uživatele z klávesnice. Při stisku klávesy pro pohyb kurzoru – šipky se *sprite* pohybuje tím směrem, kterým míří šipka. K tomu využijeme jednu z možností skupiny **Vnímání**, a to logickou funkci „klávesa [název klávesy] stisknuta?“, pomocí ní zjistíme, zda je stisknuta příslušná klávesa pro pohyb kurzoru (šipka nahoru, dolů, vlevo, či vpravo). Tuto logickou podmínku jednoduše vsuneme do místa označeného šestiúhelníkovým výřezem v příkazové kostce podmínky **Pokud**.

4 Vybíráme, či malujeme *sprite*

Jak dostat na scénu nové *sprite* a umožnit vícevláknové programování, kdy si každá postava (*sprite*) žije podle svého vlastního scénáře? Nejjednodušší je vybrat si nainstalovaný *sprite* z nabídky. Klikneme na prostřední tlačítko vedle nápisu „Nový *sprite*“ a otevře se nám dialog nabízející lidi (*people*), zvířata (*animals*), věci (*things*), písmena (*letters*), postavy pohádek a bájí (*fantasy*) a dopravní prostředky (*transport*). Mezi dopravními prostředky nechybí létající koberec, helikoptéra, či plachetnice (viz obr. 2). Vyberme si třeba zelené auto.

Obr 2: Dialog pro výběr *sprite* z nainstalovaných

K levému okraji scény umístíme auto a za něj kocoura tak, aby to vypadlo, jako když kocour

sedí v autě a řídí ho (viz obr. 3). Po kliknutí na auto se auto rozjede, přejede celou scénu a téměř zmizí za jejím okrajem (viz obr. 4). Kocour si toho všimne v okamžiku, kdy auto dorazí k okraji scény, a rozběhne se za ním. Ve skutečnosti ovšem využijeme rozesílání zpráv. Auto v okamžiku, kdy se dotkne okraje scény rozešle zprávu „jedu“ (zpráva se rozešle všem *sprite*ům na scéně, ale jen některé na ni mají naprogramovanou reakci). Kocour se ihned po obdržení zprávy „jedu“ (viz. obr. 5) rozběhne za autem, ale když „vidí“, že už ho nedohoní, zastaví zhruba uprostřed scény.

Obr 3: Scénář s kocourem dohánějícím auto

Obr 4: Skript pro pohyb zeleného auta

Obr 5: Skript pro pohyb kocoura

Nové *sprite* je možné kreslit v jednoduchém editoru obrázků přímo ve vývojovém prostředí SCRATCH, nebo je možné je vytvořit v libovolném grafickém editoru, který umožní

ukládání ve formátu PNG, nebo GIF s průhlednou barvou. Důležité je uložit nakreslené obrázky do vhodné složky ve struktuře instalačních složek programu (při standardní instalaci prostředí SCRATCH jde o složku C:\Program Files (x86)\Scratch\Media\Costumes).

Pokud nemáme výtvarné nadání, můžeme si vyhrát i s jednoduchými geometrickými tvary. V následujícím příkladu využijeme kruhové výseče (přesněji řečeno čtvrtkruhy – viz obr. 6) různých barev (modrá, červená, světle zelená a okrová) a vyzkoušíme si, jak funguje rozesílání zpráv pro větší počet spritů.

Obr 6: Scénář s kocourem a rotujícím míčem

Po kliknutí na sprite kocoura se vyšle zpráva „roztužte“. Na tuto zprávu reagují všechny čtvrtkruhy otočením o 360° (viz obr. 7). Protože střed otáčení čtvrtkruhů je umístěn ve středu kruhu, z kterého vznikly, působí celek jako rotující míč.

Obr 7: Skript pro rotaci jedné kruhové výseče

Stejný princip rotace a posílání zpráv je využit u permutačního hlavolamu, který naprogramoval v prostředí SCRATCH autor tohoto článku. Na komunitním webu jej najdete pod označením *mim3permutationteaser*.

K vytváření hlavolamů složených s dílků je vývojové prostředí projektu SCRATCH ideálním nástrojem, protože grafické prvky představující dílky lze snadno nakreslit v různých bitmapových grafických editorech, s využitím průhledné barvy

jim lze dát rozmanité tvary a pomocí komunikace mezi sprity koordinovat jejich vzájemné pohyby.

Obr 8: Úvodní instrukce scénáře - kocour vybízí k zamíchání permutačního hlavolamu

Permutační hlavolam umí jednotlivé kruhové výseče náhodně promíchat (míchání se spustí kliknutím na sprite kocoura – viz obr. 8), takže výchozí pozice je pokaždé jiná (příklad výchozí pozice lze vidět na obr. 9). Úkolem řešitele je seřadit barvy výsečí tak, jak naznačují barevné proužky nad oblouky ložisek. Povolené tahy jsou vyvolány stiskem následujících numerických kláves:

- klávesa „1“ znamená rotaci levého kruhu proti směru hodinových ručiček,
- klávesa „2“ znamená rotaci kruhu uprostřed hlavolamu proti směru hodinových ručiček, tento kruh vznikne při vzájemném posunu ložisek, při kterém současně dočasně zanikne levý a pravý kruh,
- klávesa „3“ znamená rotaci pravého kruhu proti směru hodinových ručiček,
- klávesa „4“ vyvolá vzájemný posun ložisek, a to horního vlevo a spodního vpravo,
- klávesa „6“ vyvolá vzájemný posun ložisek, a to horního vpravo a spodního vlevo,
- klávesa „7“ znamená rotaci levého kruhu ve směru hodinových ručiček,
- klávesa „8“ znamená rotaci kruhu uprostřed hlavolamu ve směru hodinových ručiček, tento kruh vznikne při vzájemném posunu ložisek, při kterém současně dočasně zanikne levý a pravý kruh,
- klávesa „9“ znamená rotaci pravého kruhu ve směru hodinových ručiček.

Vysvětlováním jednotlivých skriptů patřících prvkům permutačního hlavolamu by se tento článek neúměrně prodlužoval. Přitom kompletní kód všech skriptů je veřejně přístupný, jak je

v projektu SCRATCH zvykem, na webu <http://scratch.mit.edu/>. Stačí do vyhledávacího okénka napsat řetězec „mim3permutationteaser“, vyhledat příslušný program, načíst si jej do své instalace vývojového prostředí a kód skriptů prostudovat.

Obr 9: Pokračování scénáře – hlavolam je již zamíchán a kocour vybízí k jeho řešení

5 Závěr

Tento stručný přehled vlastností moderního dětského programovacího jazyka SCRATCH může někomu připadat nesystematický, snad i trochu chaotický. To je ale typický atribut jazyka SCRATCH, jehož název odkazuje na diskžokeji používané skrečování, tedy na první pohled chaotické, trhavé pohyby gramofonovou deskou, při nichž jehla nesnímá originální záznam přesně, ale zkresleně, útržkovitě a z těchto útržků skládá novou hudební kvalitu.

Podobným způsobem skládají programátoři skript nejen z připravených kostek, ale také z útržků skriptů kolegů z komunity, ze spritů, které našli v instalaci, nebo sami vytvořili, z kousků fotografií, které sami vyfotografovali. Někoho přilákají chytré algoritmy, jiného efektní obrázky, či zvuky. Je to programování? V každém případě zde vytvářejí malí i větší programátoři své vlastní skripty, své animace, příběhy, hry, nebo hlavolamy. Konstruují nejen

konkrétní program, ale vytvářejí si ve své mysli konkrétní představu pojmů algoritmus a program. O výsledky své práce se dělí s ostatními členy komunity. To jsou bezpochyby cenné přínosy projektu SCRATCH.

6 Literatura

- [1] COLLECTIVE. *Getting started with SCRATCH version 1.4*. 1st ed. Lifelong Kindergarten Group, MIT Media Lab, Cambridge (Massachusetts): 2009. 14 p.
- [2] MÖNIG, Jens – HARVEY, Brian. *BYOB 3.1 – Build Your Own Blocks* [online]. Berkeley University of California. [cit. 2012-09-01] Dostupné z: <<http://byob.berkeley.edu/>>
- [3] LE COQ, L. *XLOGO: Reference Manual*. 1st ed. XLOGO Comunity: 2009. Translation: G. Walker. 145 p.
- [4] HALOUSKOVÁ, Alena. *Učebnice programovacího jazyka Scratch* [online]. Brno, 2013 [cit. 2013-02-18]. Diplomová práce. MU, Fakulta informatiky. Vedoucí práce Tomáš Pitner. Dostupné z: <http://is.muni.cz/th/346458/fi_m/>.
- [5] RESNICK, Mitchel et al. Scratch: Programming for All. *Communications of the ACM*, November 2009, Vol. 52 No. 11, p. 60 – 67, DOI: 10.1145/1592761.1592779
- [6] KOLEKTIV. *SCRATCH – vymysli – programuj – poděl se* [online]. Lifelong Kindergarten Group at the MIT Media Lab. [cit. 2012-09-01] Dostupné z: <<http://scratch.mit.edu/>>
- [7] HALOUSKOVÁ, Alena. *Učebnice jazyka Scratch* [online]. Praha, 2012 [cit. 2013-02-18]. Dostupné z: <<http://perun.mff.cuni.cz/ucebnicejazykascratch/>>.

PhDr. Michal Musílek, Ph.D.

Katedra informatiky

Přírodovědecká fakulta UHK

Rokitanského 62

500 03, Hradec Králové, ČR

Tel: +420 493 332 403

E-mail: michal.musilek@uhk.cz

Www pracoviště: www.uhk.cz