

OBJEKTOVĚ ORIENTOVANÉ PROGRAMOVÁNÍ ROBOTICKÉ KOULE ORBOTIX SPHERO

LAVRINČÍK Jan, CZ

Resumé

Článek se zabývá problematikou objektově orientovaného programování v aplikacích MacroLab a orbBasic robotické koule Orbotix Sphero. Pozornost zaměřujeme zejména na možnosti vizuální a pohybové demonstrace základních algoritmů, pohybu, prostorové orientace, cyklů a Timeru. V závěru příspěvku se polemizujeme nad využitím ve výuce algoritmizace a programování v rámci vyššího sekundárního vzdělávání.

Klíčová slova: objektově orientované programování, Orbotix Sphero, MacroLab, orbBasic.

OBJECT-ORIENTED PROGRAMMING ROBOTIC BALL ORBOTIX SPHERO

Abstract

The article deals with object-oriented programming applications MacroLab orbBasic a robotic ball Orbotix sphere. Focusing in particular on the possibility of visual and physical demonstrations of basic algorithms, movement, spatial orientation, and the timer cycles. In the conclusion, the polemic over the use in the teaching of algorithms and programming in upper secondary education.

Key words: object-oriented programming, Orbotix Sphero, MacroLab, orbBasic.

Úvod

V současné době se pozornost v oblasti ICT přesunuje z běžných počítačů na menší zařízení typu počítačových tabletů (1), chytrých mobilních telefonů, hodinek atp. Tato zařízení díky mobilitě a bezdrátové konektivitě, která zahrnuje sítě WiFi, 3G, LTE, bluetooth nabízí prostor pro rozšíření zařízení o další funkce prostřednictvím bezdrátově připojeného externího zařízení. Naším cílem je analyzovat robotická zařízení s potenciálem využití ve vzdělávání. Pod pojmem „potenciál ve vzdělávání“ si představujeme možnost správy technických parametrů a nastavení, objektově orientovaného programování s možností dalšího rozšíření ve speciálním software.

1 Objektově orientované programování

Jednou z možností, jak přistoupit k problematice programování ve výuce programování v rámci sekundárního vzdělávání je využití standardních programovacích prostředí. Pod operačními systémy Windows je nejrozšířenější prostředí Visual Studio .NET (2). V rámci něj najdeme aplikaci Visual Basic, která je založena na paradigmatu objektově orientovaného programování (OOP). Koncept OOP je práce s objekty, skládání objektů, delegování služeb, dědičnost a polymorfismus (3). Jako příklad OOP prostředí můžeme uvést programovací jazyky Perl, Smalltalk, Java, C++, Object Pascal, C#, Visual Basic .NET, Lisp, PHP, Python a Ruby. Tyto jazyky spojuje stejná koncepce a nutná znalost a orientace v profesionálním programovacím prostředí. Pro žáky v rámci sekundárního vzdělávání není vždy orientace v tomto prostředí snadná (4), (5). K problematice můžeme přistoupit vytvořením nadstavby v prostředí Visual Basic (želví grafika – class modul). Další možností je využití potenciálu robotických zařízení, které nabízejí sofistikovaný software jako v případě robotické koule Sphero nabízející možnost přístupu k objektově orientovanému programování formou nastavování parametrů vlastností a metod vytvořených objektů.

2 Programování Orbotix Sphero

Společnost Orbotix nabízí dva druhy software určeného k programování své robotické koule. MacroLab nabízí již vytvořené objekty pro pohyb, otáčení, barvy, časování a další funkce koule. OrbBasic je prostředí, ve kterém nejsou vizuální hotové objekty a vše se zapisuje ve formě zdrojových kódů (6), (7). Některé z prvků již můžeme znát s programovacími prostředími typu Imagine Logo (želví grafika). V prvním příkladu se podíváme na pohyb ve tvaru čtverce.

Příklad 1: Pohyb ve čtverci (každá strana = jiná barva).

RGB 1.0 0.0 0.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % červené složky, 100 = zdržení 100 ms

Roll 0.5 0 2000 `pohyb o rychlosti 0,5 = 50 % rychlosti, úhel vůči původní pozici koule 0°, zdržení 2000 ms

RGB 0.0 1.0 0.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % zelené složky, 100 = zdržení 100 ms

Roll 0.5 90 2000 `pohyb o rychlosti 0,5 = 50 % rychlosti, úhel vůči původní pozici koule 90°, zdržení 2000 ms

RGB 0.0 0.0 1.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % modré složky, 100 = zdržení 100 ms

Roll 0.5 180 2000 `pohyb o rychlosti 0,5 = 50 % rychlosti, úhel vůči původní pozici koule 180°, zdržení 2000 ms

RGB 1.0 1.0 0.0 100 `změna barvy dle barevného modelu RGB 1.0 1.0 = 100 % modré složky a 100 % zelené složky, 100 = zdržení 100 ms

Roll 0.5 270 2000 `pohyb o rychlosti 0,5 = 50 % rychlosti, úhel vůči původní pozici koule 270°, zdržení 2000 ms

Druhý příklad je využití integrovaných LED diod a možnosti cyklů. V prostředí MacroLab nejsou k dispozici klasické cykly typu for – next nebo do - loop, které známe z moderních programovacích jazyků typů Visual Basic .NET, ale pouze modifikace těchto typů cyklů dohromady.

Příklad 2: Cyklická změna barev.

Loop start 9 `parametr počet opakování 9

RGB 1.0 1.0 1.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % zelené, modré i červené složky, 100 = zdržení 100 ms (výsledná barva bílá)

RGB 0.0 0.0 1.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % modré složky, 100 = zdržení 100 ms (výsledná barva modrá)

RGB 0.0 1.0 0.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % zelené složky, 100 = zdržení 100 ms (výsledná barva zelená)

RGB 1.0 0.0 1.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % modré i červené složky, 100 = zdržení 100 ms (výsledná barva růžová)

RGB 1.0 0.0 0.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % červené složky, 100 = zdržení 100 ms (výsledná barva červená)

Loop End `konec cyklu

Ze speciálních funkcí můžeme zmínit efekt fade umožňující zpomalený náběh barev nebo plynulý přechod barev. Při jeho použití používáme kombinaci objektů RGB a Fade, případně práci s časovačem delay. Pro opakování efektu se taktéž nabízí kombinace s cyklem loop. Z důvodu umístění základní desky a baterie nesvítilí zařízení ze všech stran stejně (zejména při použití

průhledné Apple verze Sphero 2). Pro eliminaci tohoto jevu se nabízí použít možnost příkazu rotate 360, který nechá kouli rotovat o úhel 360° na místě.

Příklad 3: Rotace na místě se změnou barvy.

```
Loop Start 3 `parametr počet opakování 3
Back LED 1.0.0 `rozsvícení kalibrační zpětné diody
RGB 1.0 1.0 0.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % červené složky, 100 =
zdržení 100 ms (výsledná barva červená)
Rotate 360 1000 `rotace o úhel 360° v délce trvání 1000 ms
Delay 1000 `vyčkání 1000 ms
RGB 1.0 0.0 1.0 100 `změna barvy dle barevného modelu RGB 1.0 = 100 % modré i červené
složky, 100 = zdržení 100 ms (výsledná barva růžová)
Rotate -360 1000 `rotace o úhel 360° v délce trvání 1000 ms
Loop End `konec cyklu
```

Z dalších méně používaných, ale užitečných funkcí můžeme jmenovat sleep (uspání zařízení) a rotation rate (rychlost otáčení). Celý set objektů je konstruován pro využití maximálního potenciálu zařízení Sphero a nabízí možnost vytvoření mnoha zadání pro žáky např. opisujte pomocí Sphero pravidelný čtverec o délce strany 1 metr, který žák musí dopočítat z parametrů rychlosti a času operace v milisekundách).

Závěr

Robotická koule Orbotix Sphero nabízí jako přidanou hodnotu možnost objektově orientovaného programování prostřednictvím aplikací MacroLab a orbBasic (8), (9). Jejich využití můžeme díky nízké pořizovací ceně za hardware a bezplatnému softwaru doporučit pro výuku objektového programování v rámci nižšího nebo vyššího sekundárního vzdělávání. Programování nabízí několik úrovní pokročilosti, přičemž v něm lze pokračovat i v rámci terciálního vzdělávání využitím API Orbotix pro vývoj vlastních aplikací s využitím robotické koule Orbotix Sphero.

Literatura

1. STAMAKAKIS, W. *Microsoft Visual Basic Design Patterns*. 1st edition. Washington: Redmont. 262 p. ISBN 978-1572319578.
2. GRUNDGEIGER, D. *CDO & MAPI Programming with Visual Basic*. 1st edition. USA: O'Reilly, 2000. 384 p. ISBN 1-56592-665-X.
3. KLEMENT, M. *Základy programování v jazyce Visual Basic*. 1. vyd. Olomouc: VUP, 2002. 336 s. ISBN 80-244-0472-9.
4. KLEMENT, M. *Základy programování v jazyce Visual Basic 1: Úvod do MS Visual Basic 6.0 (studijní opora pro kombinované studium)*. 1. vyd. Olomouc: VUP, 2008. [CD-ROM]. ISBN 978-80-244-2177-3.
5. KLEMENT, M. *Základy programování v jazyce Visual Basic 2: Začátky programování v MS Visual Basic 6.0 (studijní opora pro kombinované studium)*. 1. vyd. Olomouc: VUP, 2008. [CD-ROM]. ISBN 978-80-244-2178-0.
6. KLEMENT, M. *Základy programování v jazyce Visual Basic 3: Pokročilejší programování v MS Visual Basic 6.0 (studijní opora pro kombinované studium)*. 1. vyd. Olomouc: VUP, 2008. [CD-ROM]. ISBN 978-80-244-2179-7.
7. HAVELKA, M. Uplatnění konstruktivistických přístupů v přípravě učitelů technické a informační výchovy ve výuce oborové didaktiky. In CHRÁSKA, M. KLEMENT, M.

SERAFÍN, Č. HAVELKA, M. *Trendy ve vzdělávání 2011*. Olomouc: agentura gevak s.r.o., 2011, s. 61-65. ISBN 978-80-86768-34-2.

8. SCHELTEN A. *Konstruktivistische Lernauffassung und Hochschullehre* [online], [cit. 2010-03-02]. Dostupné na URL: <<http://www.paed.ws.tum.de/>>.
9. DOSTÁL, J. Multimediální, hypertextové a hypermediální učební pomůcky – trend soudobého vzdělávání. *Journal of Technology and Information Education*. 2009, Olomouc - EU, Univerzita Palackého, Ročník 1, Číslo 2, s. 18 - 23. ISSN 1803-537X (print). ISSN 1803-6805 (on-line).

Lektorovali: Mgr. Zdeňka Křišová, Mgr. Martin Havelka, Ph.D.,

Kontaktní adresa:

Jan Lavrinčík, PhDr. DiS.,

Katedra technické a informační výchovy, Pedagogická fakulta UP, Žižkovo nám. 5, 771 40 Olomouc, ČR,
tel.: 00420 585 635 8011, fax +420 585 231 400, e-mail: nobilis.felis@seznam.cz