

Programování virtuálních agentů – Platforma Pogamut

RUDOLF KADLEC, JAKUB GEMROT, CYRIL BROM

Kabinet softwarové a výuky informatiky, MFF UK Praha

Malostranské nám. 25, 118 00 Praha

{rudolf.kadlec, jakub.gemrot}@gmail.com, brom@ksvi.mff.cuni.cz

Abstrakt. Tento tutoriál se zabývá programováním umělé inteligence virtuálních postav v počítačových hrách, tzv. virtuálních agentů nebo též botů. Vybrané techniky budou prezentovány na softwarové platformě Pogamut.

Klíčová slova: virtuální agenti, boti, programovací nástroje.

1 Úvod

Pojem *virtuálním agent* („virtual agent“) označuje počítačový program, který imituje lidské nebo zvířecí chování ve virtuálním prostředí a který disponuje virtuálním tělem, prostřednictvím něž vnímá a jedná. Důraz je kladen na existenci virtuálního těla jeden z hlavních definujících prvků této podskupiny *inteligentních softwarových agentů* („intelligent software agent“) [19]. Virtuálním prostředím se rozumí určitá abstrakce našeho přirozeného reálného světa. Tato abstrakce je uživateli typicky prezentována v 3D zobrazení, jaké známe například z akčních počítačových her. Pro prototypování se ale často používají i 2D světy. V kontextu počítačových her jsou virtuální agenti často nazýváni *boti* („bots“). Termín boti budeme používat i v tomto textu.

Vytvořit věrohodného bota znamená vyřešit problémy spadající do různých vědních disciplín; počítačové graficky, fyzikálního modelování, psychologie ad. Zde se budeme zabývat pouze umělou inteligencí botů, čili problematikou jejich *řízení*. Tvorba algoritmů řídicích chování botů byla až donedávna výsadou expertů. V poslední době se ale objevily nástroje zpřístupňující tuto oblast i méně zkušeným programátorům [7]. Příkladem takového nástroje je i platforma Pogamut [6] použitá v rámci tohoto tutoriálu i při výuce na naší univerzitě [3]. První verze platformy byla vyvíjena na MFF UK v letech 2006-7. V současné době je k dispozici verze 3 [14].

Díky Pogamutu je programování botů přístupné i pro začínající programátory, stále ale zůstávají oblasti, s nimiž musí být začátečník seznámen předtím, než začne vytvářet svého prvního bota. V tomto tutoriálu popíšeme tyto třídy úloh a zároveň je budeme demonstrovat na praktické ukázce programování středně složitě bota. Rozšiřující materiály lze najít na [1, 2, 4, 5].

2 Základy programování botů

Skupině algoritmů a datových struktur řídicí chování bota se často říká *umělá mysl* („artificial mind“). Ta má typicky modulární architekturu. Tři základní moduly, jimiž je vybaven téměř každý bot, jsou modul pro vysokoúrovňové plánování akcí, pro dlouhodobé

plánování cesty a pro řešení vyhýbání se překážkám. Budeme se jimi zabývat i v tomto textu a dále pak důkladněji během tutoriálu.

Modul vysokoúrovňového rozhodování (v angličtině často nazývaný „decision making system“ – DMS – nebo „action selection mechanism“ – ASM) je centrálním arbitrem koordinujícím práci dalších modulů. Prostředí nabízí botovi omezenou množinu akcí a úkolem DMS je vybrat v aktuálním okamžiku tu nejvhodnější. Na konceptuální úrovni se pro popis DMS často používají konečné automaty nebo reaktivní pravidla (často také označovaná jako IF-THEN pravidla). Podoba s pravidly z expertních systémů není příliš velká: DMS typicky neprovádí inferenci způsobem, na jaký jsme z jiných oblastí umělé inteligence zvyklí. Důvodem je výpočetní náročnost; v oblasti umělé inteligence pro počítačové hry se i kvadratická složitost algoritmu obvykle hodnotí jako příliš vysoká. Podobně analogie mezi konečnými automaty řídícími boty a konečnými automaty používanými v jiných partiích informatiky je spíše na povrchové úrovni.

Úkolem modulu plánování cesty je výpočetně efektivní nalezení cesty mezi botovou výchozí pozicí a cílovou lokací. Pro tyto účely je často používán algoritmus A* [16], heuristický algoritmus na prohledávání grafů, nebo jeho odvozeniny. Grafy, které abstrahují 3D (2D) svět pro potřeby plánovače, jsou typicky zadány předem. Existují ale i algoritmy pro jejich automatické generování.

Poté, co je cesta naplánována, může ji bot začít následovat. V dynamických prostředích nemusí ale bezmyšlenkovitě následování předem vypočítané cesty fungovat. Pohyblivé objekty mohou například cestu zatarasit. Naštěstí existuje několik jednoduchých a výpočetně nenáročných algoritmů, které dokáží problém lokálně blokované cesty vyřešit. Většina těchto algoritmů je odvozena od mechanismů *lokálního řízení* („steering“) popsanych Reynoldsem [15]. V průběhu tutoriálu popíšeme jak algoritmus A* tak mechanismy lokálního řízení a ukážeme, jak mohou být obě tyto techniky integrovány se zbytkem DMS.

Pokud bot, kterého vytváříme, má imitovat lidské chování, začnou získávat na důležitosti i další témata – například umělé emoce. Emocionální odezvy na události v botově okolí můžeme buď předem naskriptovat, nebo je můžeme generovat obecným modelem vzniku emocí, jakým je například ALMA [8]. Díky zapojení obecného modelu mohou vznikat emergentní jevy, které programátor původně ani nezamýšlel (což má svá pozitiva i negativa). Téma emocionálního modelování se bude věnovat v poslední část tohoto tutoriálu.

3 Platforma Pogamut

Platforma Pogamut je softwarový nástroj pro usnadnění vývoje a testování umělých myslí botů [6, 11, 14]. Platforma řeší nízkoúrovňové problémy jako je komunikace s virtuálním světem, hledání cesty nebo vyhýbání se překážkám. Umožňuje tím vývojáři plně se soustředit na tvorbu botova chování.

Platforma se skládá z těchto hlavních komponent:

- Virtuální prostředí – jako simulátor virtuálního světa je použita komerční hra Unreal Tournament 2004 rozšířená o komunikační rozhraní GameBots2004 distribuované pod licenci GNU. Obr. 1 ukazuje vizualizaci tohoto prostředí.
- Knihovna tříd – knihovna v jazyce Java poskytuje API pro programování chování botů.

Tutoriál



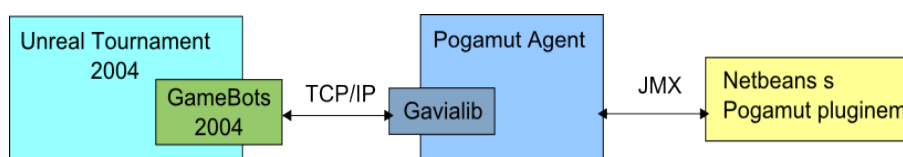
Obr. 1: Interakce postav v prostředí hry Unreal Tournament 2004 (Copyright 2009 Epic Games)

- Integrované vývojové prostředí – plug-in pro vývojové prostředí Netbeans [12] poskytuje podporu při programování bota, při jeho ladění a při provádění experimentů.
- Emoční generátor – obecný model vzniku emocí umožňuje obdařit každého bota unikátní osobností.

Architekturu platformy Pogamut ukazuje Obr. 2. Boti jsou vtěleni do prostředí hry Unreal Tournament 2004, kde jsou grafické reprezentace jejich těl ovládány pomocí komunikačního protokolu GameBots2004. To umožňuje, aby jejich řídicí logika běžela v samostatném procesu – např. programu v jazyce Java. Knihovna GaviaLib odstiňuje programátora od detailů TCP/IP komunikace textového protokolu GameBots2004. Informace o běhu agenta užitečné pro jeho ladění jsou pak zprostředkovány plug-inem pro vývojové prostředí Netbeans.

Kromě úvodu do obecných technik programování botů bude součástí tutoriálu i naprogramování ukázkového bota – Lovce („Hunter“). Na Lovci bude ukázána základní funkcionality potřebná pro akční hry, jako je navigace v prostředí, vyhýbání se překážkám, pronásledování hráčů, sbírání objektů a výběr zbraně. Všechna tato chování jsou popsána sadou IF-THEN pravidel.

Materiály popisující základní interakci s platformou Pogamut krok za krokem, vhodné i pro samostatné vyzkoušení platformy, jsou dostupné na adrese [17]. Platforma také obsahuje několik dobře okomentovaných ukázkových botů, kteří mohou sloužit jako výchozí bod pro naprogramování vlastního bota. Dokumentace projektu a online podpora na diskusních fórech je dostupná na domovské stránce [14].



Obr. 2: Architektura platformy Pogamut

4 Rozšíření

Platforma Pogamut je stále aktivně rozšiřována. Poslední rozšíření zahrnují připojení kognitivní architektury ACT-R [20], modul pro snadnou tvorbu výukových scénářů specifikovaných v omezeném přirozeném jazyce [18], podporu pro animace popsané v jazyce BML [13], vizuální editor reaktivních plánů jazyka POSH [10] a v neposlední řadě jazyk StorySpeak pro scriptování interakce více postav [9].

5 Závěr

Účelem toho textu bylo podat krátký přehled problémů spojených s tvorbou botů a zároveň představit platformu Pogamut, softwarový nástroj pro snadné prototypování chování botů. Teoretické aspekty tvorby botů i jejich programování s pomocí platformy Pogamut budou přiblíženy v průběhu tutoriálu.

Poděkování

Tato práce byla podpořena grantem “Informační Společnosti” pod projektem 1ET100300517, dále grantem 201/09/H057 a výzkumným záměrem MSM0021620838 MŠMT. Tvorba výukových scénářů pro projekt Pogamut je umožněna díky projektu CZ.2.17/3.1.00/31162, který je financován Evropským Sociálním Fondem, rozpočtem České Republiky a rozpočtem hl. m. Prahy.

Literatura

1. Brom, C.: Curricula of the course on modelling behaviour of human and animal-like agents. In: *Proceedings of the Frontiers in Science Education Research Conference*, Famagusta, North Cyprus. (2009) 71 -- 79
2. Brom, C., Bryson J. J.: White paper on „Action Selection for Intelligent Systems“, EUCognition, The European Network for the Advancement of Artificial Cognitive Systems. (2006) URL: <http://www.eucognition.org/asm-whitepaper-final-060804.pdf> [11.12.2008]
3. Brom, C., Gemrot, J., Burkert, O., Kadlec, R., Bida, M.: 3D Immersion in Virtual Agents Education In: *Proceedings of First Joint International Conference on Interactive Digital Storytelling*, ICIDS 2008 Erfurt, Germany, November 26-29, LNCS 5334, Springer-Verlag, Berlin (2008) 59-70

Tutoriál

4. Brom, C., Sisler, V., Hoffmann, M.: Virtuální lidé. Kapitola knihy. In (Mařík V., Stěpánková O., Lažanský J., eds): *Umělá inteligence 5*. Academia, Praha (2007)
5. Bryson J.: How to Make a Monkey Do Something Smart. <http://www.cs.bath.ac.uk/~jjb/web/how-to-monkey.pdf> (2001) [11.12.2008]
6. Burkert, O., Kadlec, R., Gemrot, J., Bída, M., Havlíček, J., Dörfler, M., Brom, C.: Towards fast prototyping of IVAs behavior: Pogamut 2 In: *Proceedings of 7th International Conference on Intelligent Virtual Humans*, LNCS Vol. 4722. Paris, France. Springer-Verlag, Berlin. (2007)
7. Cooper, S., Dann, W., Pausch, R.: Teaching Objects-first In Introductory Computer Science. In: *Proceedings of the SIGCSE technical symposium on Computer science education* (Reno, Nevada, USA). ACM New York, NY, USA. 191-195. (2003)
8. Gebhard, P.: ALMA - A Layered Model of Affect. In: *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)*, 29-36, Utrecht. (2005) URL: <http://www.dfki.de/~gebhard/alma/index.html> [5.6.2009]
9. Gemrot, J.: *Joint behaviour for virtual humans*. Diplomová práce. Univerzita Karlova v Praze, 2009.
10. Havlíček, J.: *Prostředky pro podporu ladění v projektu Pogamut 2*. Bakalářská práce. Univerzita Karlova v Praze, 2009.
11. Kadlec, R., Gemrot, J., Burkert, O., Bída, M., Havlíček, J., Brom, C., : POGAMUT 2 - A platform for fast development of virtual agents' behaviour, In: *Proceedings of CGAMES 07*, La Rochelle, France (2007)
12. Netbeans homepage: URL: <http://www.netbeans.org> [28.8.2009]
13. Píbil, R.: *Gesta a mimika virtualních postav v 3D světě*. Bakalářská práce. Univerzita Karlova v Praze, 2009.
14. Projekt Pogamut: URL: <http://artemis.ms.mff.cuni.cz/pogamut> [28.8.2009]
15. Reynolds, C.: Steering Behaviors for Autonomous Characters. In: *Game Developers Conference*, (1999)
16. Russell, S. J.; Norvig, P.: *Artificial Intelligence: A Modern Approach*. Upper Saddle River, N.J.: Prentice Hall. (2003) 97–104. ISBN 0-13-790395-2.
17. Tutorialy platformy Pogamut: URL: http://artemis.ms.mff.cuni.cz/pogamut_files/latest/doc/tutorials/ [28.8.2009]
18. Vansa, R.: *Výukové scénáře v projektu Pogamut*. Bakalářská práce. Univerzita Karlova v Praze, 2009.
19. Wooldridge, M.: *An Introduction to Multiagent Systems*, John Wiley & Sons, Chichester, England. (2002)
20. Zemčák, L.: *ACT-R v projektu Pogamut*. Bakalářská práce. Univerzita Karlova v Praze, 2009.

Annotation:

Programming of virtual agents – The Pogamut Platform

This paper describes the content of our tutorial concerned with the development of artificial intelligence for virtual characters in computer games, so-called bots. The Pogamut platform was chosen for this tutorial. Step-by-step manual that will be followed during this tutorial is available at http://artemis.ms.mff.cuni.cz/pogamut_files/latest/doc/tutorials/.