Možnosti a srovnání free enginů pro umělou inteligenci herních avatarů

referát do předmětu PB016 Úvod do umělé inteligence

Dan Musil
4309
xmusil2@fi.muni.cz

Vývoj a význam enginů pro umělou inteligenci

Na hry jsou kladeny čím dál větší nároky, a to nejen v oblasti 3D akcelerátorů a zvuku, ale v poslední době se opět herní vývojáři vracejí ke snaze vdechnout své hře co nejvyšší hratelnost. Po vzhledové stránce jsou hry dostatečně dokonalé, chybí jim však „život". Chování počítačových protivníků je nelogické, nedokonalé, předvídatelné. To stejné platí ve všech oborech, do kterých umělá inteligence proniká – tedy do všech. Ať už se budeme mluvit o robotice, expertních systémech, animacích, webových aplikacích nebo kuchyňských spotřebičích, použitá inteligence nedostačuje a existuje nemalý tlak i motivace ji neustále vylepšovat.

Bohudík se tak děje, dokonce mílovými kroky. Počítače jsou čím dál lépe schopny rozpoznávat lidskou řeč, analyzovat obraz nebo si poradit v krizových situacích bez lidského zásahu. Velikou mírou se na vývoji podílí univerzity, získaným poznatkům pak dají rámec komerční firmy. Stejně jako v případě grafiky, zvuku a fyziky vznikají enginy (vývojové knihovny) pro umělou inteligenci, které usnadňují začlenění prvků inteligence do libovolné aplikace, animace či zařízení. Jakmile bude ustálena použitelná sada řešení (prozatím neexistuje), vznikne jistě hardwarová podpora v podobně přídavné karty. Při řešení konkrétních malých úkolů se tak již stalo (šachy, apod).

Často řešené problémy a studium umělé inteligence

Nejjednodušší, studijně nejzajímavější a s nejrychlejšími výsledky je metoda analyzovat problém konkrétní hry, algoritmizovat řešení a navrhnout strategii pro počítač. Kroky, které vedou k vítězství. Částečně to možné je, řešení se setkává s úspěchem, mnohokrát je dokonce více než dostačující. Nicméně z hlediska vývoje umělé inteligence tato metoda tvoří slepou větev.

Algoritmy jsou sice velmi výkonné a účinné, nicméně příliš jednoúčelové, neflexibilní a zpravidla řešené výpočetní silou či obrovskou databází. Žádné náznaky učení, emocí, analýzy.

V dnešní době už jen ke studijním účelům jsou určeny následující hry: Mastermind, Piškvorky, Othello (Reversi), Dáma, Backgammon, Šachy, Scrabble, Go, Bridge, Poker. Dle schopností implementátora jsou jednotlivé aplikace na VELMI rozdílné úrovni, ovšem také složitosti.

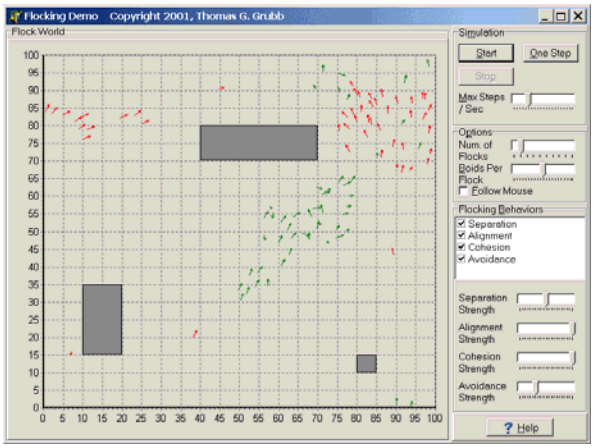
Velmi zajímavé ke studiu jsou hry, které umožňují pomocí různých zjednodušených jazyků naskriptovat si chování vlastního agenta. Z hlediska umělé inteligence to velký význam nemá, ale naučíte se při tom, jak myslet při programování agenta. Zkuste například následující: AI wars, Bolo, Terrarium, Half-life, Quake, Red Alert, Unreal a další.

Řešené problémy, situace

Jelikož používám umělou inteligenci nejen k provozování svého koníčku, čímž je programování počítačových her, ale používáme ji i při řešení běžných úloh v profesionálním vývoji aplikací (CRM, webové aplikace), snažil jsem se analyzovat trh a možnosti současných balíků pro umělou inteligenci. Nenašel jsem žádnou, která by se snažila o zahrnutí všech známých a dostupných algoritmů. Snad to ani není cílem. Mnohokráte tyto balíky nebyly dokončeny, chyběla jim dokumentace nebo dokonce nešly zkompilovat (oficiálně potvrzeno).

Po zkoumání jsem sepsal několik oblastí, které enginy obvykle řeší. Někdy jednu, občas i dvě, výjimečně i tři z nich. Některé ze skupin dokáží nahradit elementárně skupinu jinou, o výhodnosti však lze dlouhosáhle diskutovat.

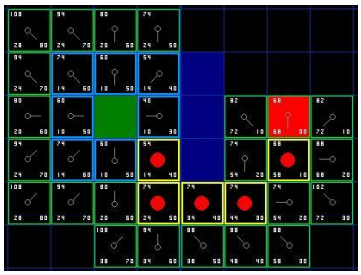
A-Life a Flocking



Enginy zahrnuté do této kategorie řeší simulaci života jako takového, flocking pak řeší problém chování skupiny, stáda. Algoritmy se snaží hráče obklopit co možná nejpřirozenějším chováním avatarů v jeho okolů. Od postav až po motýly.

Při širším pojetí je možné tuto technologii použít i k ovládání technologických celků, které spolu mají spolupracovat. US Air Force plánovala použít A-Life k ovládání satelitů, které by tak dokázaly spolupracovat a inteligentně se doplňovat tak, aby určenou úlohu dokázaly splnit co nejefektivněji.

Pathfinding



Pravděpodobně nejjednodušší a také nejoblíbenější část umělé inteligence. Jasný cíl, poměrně málo nepřekonatelných problémů. Stále však zůstává problémem, protože implementace je příliš specifická pro daný problém.

K tomuto problému není problém najít obrovské množství zdrojů, algoritmů (A*, D*, Dijkstra) a teorií. Důsledný pathfinding je značně náročný na výpočetní výkon. Zajimavá je rovněž implementace vyhledávání nejefektivnější cesty v prostoru.

Genetické algoritmy

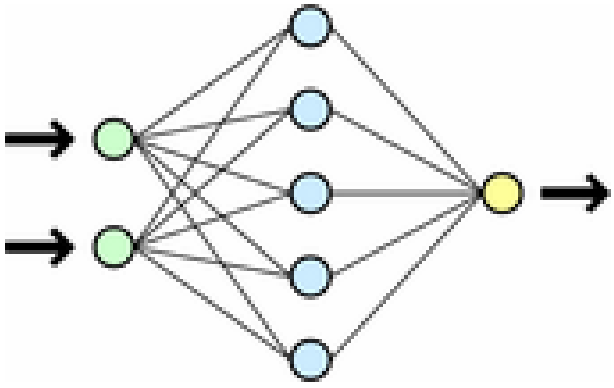
Umí řešit řadu úloh. I ty, pro které máme rychlejší a efektivnější řešení (vyhledávání cesty pomocí A*). Spolu s neuronovýma sítěm a fuzzy logikou tvoří nejpoužívanější technologie v umělé inteligenci v současnosti. Je to především obrovskou flexibilitou a obecností, kterou nabízejí. Dokáží se vyvíjet, adaptovat a inteligenci neprojevují předprogramovaných chováním. Svým způsobem se pomocí příkazů snaží napodobit chvání přírody, včetně její náhodnosti.

Jsou však genetické alogritmy použitelné i ve hrách? Obtížně se implementují, pro neznalé nejsou příliš transparentní, špatně se odhalují chyby, mají velmi široký záběr, výsledky nejsou zcela přesvědčující a implementované API přistupuje k řešení velmi různě. Jsem si jistý a pevně doufám, že i tato cesta bude rozvíjena i nadále, i kdyby se nakonec ukázala jakou slepá. Potenciál genetické algoritmy jistě mají. Kdo by si nechtěl zahrát na boha a dívat se, jak se jeho výtvory vyvíjejí a učí?

Velmi časté v enginech je spojení genetických algoritmů s umělým životem (A-Life).

Neuronové sítě

Jedná se o nejsložitější a nejkomplexnější nástroj k řešení problémů. Využívá naučenou znalostní vázby a stejně jako v mozku synapse si vytváří vazby mezi poznatky, které učinil. Používá se především v kybernetice a je značně blízký Prologu.



Ve své podstatě se matematika snaží kopírovat přírodu. Základní složkou neuronové sítě je neuron. Neuron má několik výstupů (může mít jenom jeden v případě vstupních neuronů) a několik vstupů (nebo jen jeden v případě výstupních neuronů). Tvoří některou funkci mezi těmito vstupy a výstupy. Problémem je, že funkce nebyla určena přírodním neuronem. Zjednodušme tuto neuronovou funkci tak, aby byla co nejjednodušší a bylo možné ji zrealizovat na standardním osobním počítači.

Termín neuronová síť zahrnuje spojení několika neuronů s výše zmíněnými vlastnostmi. Komplexní funkce neuronové sítě je odvozena funkcemi jednoduchých neuronů, upravením jejich výstupů a individuálním způsobem organizace spojujícího neuronu. Čím bude funkce realizovaná neuronem jednodušší, tím vyšší bude muset být počet neuronů, řečeno jiným způsobem, pokud by neuronové funkce odpovídaly síťovým funkcím, stačil by jeden. Vhodným seskupení neuronů můžeme ušetřit mnoho neuronů nebo realizovat paměť (pomocí zpětné vazby).

Fuzzy logika

Fuzzy logika se poprvé objevila v roce 1965 v článku, jehož autorem byl profesor Lotfi A. Zadeh. Tehdy byl definován základní pojem fuzzy logiky a to fuzzy množina. Slovo fuzzy znamená neostrý, matný, mlhavý, neurčitý, vágní. Odpovídá tomu i to, čím se fuzzy teorie zabývá: snaží se pokrýt realitu v její nepřesnosti a neurčitosti.

V klasické teorii množin prvek do množiny buďto patří (úplné členství v množině) nebo nepatří (žádné členství v množině). Fuzzy množina je množina, která kromě úplného nebo žádného členství připouští i členství částečné. To znamená, že prvek patří do množiny s jistou pravděpodobností (stupeň příslušnosti). Funkce, která každému prvku universa přiřadí stupeň příslušnosti se nazývá funkce příslušnosti.

Proč je vlastně fuzzy logika tak důležitá? Jednak je potřeba pracovat s vágními daty a jednak používání přesných popisů nás vede k idealizování skutečností reálného světa a tedy k odklonu od reality.

Striktní popis vede k popisu skutečnosti pouze pomocí dvouprvkové množiny {0,1}. Pokud problém nelze jednoznačně určit, rozkládá se na menší podproblémy, ale za cenu místa a opět lze použít jen dvouprvkovou množinu. V případech, kdy je již nemožné nebo neúnosné takto problém rozdělit, dopouštíme se jisté chyby a tím je dán odklon od reality.

Stavové automaty a agenti

Díky své jednoduchosti jde o nejstarší a také nejspolehlivější technologii a nějakou dobu jí nějspíše ještě zůstane. Je snadno pochopitelná, použitelná, osvědčená a dobře se hledají chyby. Na druhou stranu je hrám často vytýkána a hráči strategii brzy odhalí. Je dobré ji tedy kombinovat s nějakou další technologií.

Používá se nejčastěji k implementaci botů, chatovacích robotů a inteligentních agentů ve známém prostředí, se známými pravidly a jasným cílem. Jde o jednoduchý model rozhodování v předem definovaných situacích.

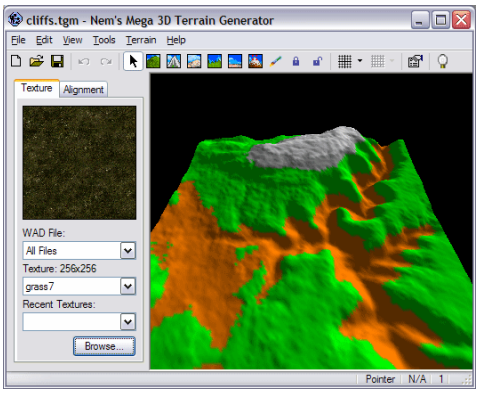
Robotika

Oproti klasickému modelu rozhodování mustí řešit analýzu obrazu, hlasu, doteků. Musí umět vyhodnocovávat a zpracovávat nečekané a nepředvídatelné situace. Je naprogramován si v takových krizových situacích „nějak“ poradit, a poté výsledek vyhodnotit a poučit se z něj.

Hodí se do mnoha oblastí života – výroba, průzkum, bezpečnost, domácnost. Poskytuje největší prostor pro teoretický výzkum a je důležitá i pro herní vývojáře. Ti si tak blíže k realitě mohou

vyzkoušet robustnost svých algoritmů. Naopak poznatky z robotiky lze aplikovat zpět do virtuální reality (ať už v jakékoli formě).
Pořádají se zajímavé soutěže robotů - http://robots.net/ .

Generování povrchů



V posledních letech se vyrojila řada programů generujících povrchy (terény). Ty, co jsem zkoušel, mi připadají příliš jednoduché a omezené. Z diskuzních skupin herních vývojářů vím, že tento problém nenesu sám. Snahou je vytvořit co nejvěrnější povrch s možností definovat jasná pravidla (umístění budov, cest, polohy důležitých prvků).
Aplikace napodobují hory, vodu, vegetaci, faunu – vše musí být přirozené a logické. Často se vychází z 2D mapy, což problematiku zjednodušuje a značně znevýhodňuje (jeskyně, převisy, anomálie v rostlinstvu).

Free enginy

Obvykle se jedná o univezitní vývoj. Problematika je od určité smysluplné úrovně pro normální lidi příliš komplikovaná. Čas na ni najdou zpravidla jen studenti nebo odborníci, kteří enginy vyvíjejí profesionálně (k těm zběžně později).
Na rozdíl od 3D, zvukových a fyzikálních enginů zatím neexistuje žádný obecný standard, skupina poskytovaných služeb, zkrátka něco, v čem jsou si enginy podobné. Jejich implementace i úroveň se značně liší. Často jsou příliš úzce zaměřené a je těžké najít takový, který mně vyhovuje. Těch komplexnějších existuje opravdu málo, leckdy jsou ve fázi návrhu, přípravy, testů. Nedodělané a v praktickém nasazení nepoužitelné. V takových případech nezbývá než licencovat některý z drahých komerčních enginů.
Implementace free enginů probíhá v různých jazycích a na různých platformách (vzpoměňte studijní účely).

Braniac

- http://www.twilightminds.com/bbe.html
- poměrně rozsáhlý engine
- skripty, stavové automaty
- ideální na programování botů a RPG
- Licence: opensource

CobaltAI

- http://www.cobaltai.com/products.htm
- v poslední době asi nejpoužívanější engine
- neuronové sítě, genetické aloritmy, fuzzy logika
- VB, C++, .Net
- použitelné v mnoha oblastech
- dobrá podpora a dokumentace
- Licence: pro osobní a pro komerční použití

DirectIA

- http://www.masa-sci.com/

- cílem je vyvinout autonomní agenty (i skupiny) se schopností učit se a reagovat v reálném čase
- chování, cíle, motivace
- fuzzy logika, stavové automaty, vlastní skriptovací jazyk (využívá pravidla)
- obsahuje nástroje pro ladění
- bohužel neobsahuje některé základní algoritmy (pathfinding)

GALib

- http://lancet.mit.edu/ga/
- univerzitní projekt (MIT)
- sada C++ tříd implementující všechny genetické algoritmy
- multiplatformní, dobře dokumentované, skvělé ke studiu
- Licence: zdarma pro osobní užití, povolení pro užití ve hře

KDCalc

- http://www.knowledgedynamics.com/kdCalcProductPage.htm
- velmi zajímavý nápad – implementace pomocných nástrojů pro AI v Excelu
- fuzzy logika, neuronové sítě, sady pravidel
- vnitřek zůstává před návrhářem AI skryt

Logic Programming Associates

- http://www.lpa.co.uk/
- skupina lidí nabízející řadu nástrojů pro vývoj AI (řadu z nich včetně grafické nadstavby)
- fuzzy logika, agenti (vlastní DB jazyk), rozhraní pro svázání AI struktur

Memetic

- Memetic (http://www.memeticai.org/)
- rozsáhlé API s vynikající dokumentací
- snaží se řídit chování objektů – dokáže ovládat NPC (chování, cíle, pocity, …), včetně interakce s hráčem
- bohužel byl použit jazyk NWScript (Neverwinter nights), který značně omezuje použití

Zajímavé komerční enginy

Tvůrci počítačových her si často implementují umělou inteligenci vlastními silami. Specificky. Zpravidla je takový kód rychlejší a v dnešních hrách bojujeme o každý výpočetní takt procesoru. Přesto (a díky za to) začínají pomalu vznikat univerzální balíky, které jsou pomůckou nejen herním vývojářům, ale i animátorům a robotikům.
Za příslušnou cenu tak vývojáři dostanou balík s vynikající podporou, vyvíjený profesionály. Balík má široké pokrytí funkcí a zpracovávaných technologií, obrovský rozsah podporovaného hardware (čím dále důležitější jsou herní konzole), jsou propracovanější a hlavně dokončené a ihned použitelné. Navíc zpravidla nabízejí přehledné klikací GUI včetně debuggeru, ve kterém si vývojář může ihned chování svých agentů vyzkoušet. Vše je přehledné a názorné – neocenitelná pomůcka, ať si říká kdo chce co chce. Jejich cena tomu odpovídá.
Často existují i jako rozšíření do 3D modelovacích nástrojů (Max, Maya), kde řídí chování objektů v animaci (skupiny, jednotlivci, reakce).

PathEngine

- http://www.pathengine.com/
- knihovna rychlých a výkonných algoritmů pro hledání cesty, a to včetně pohybujících se překážek
- obsahuje i kolizní model

Renderware AI

- http://www.renderware.com/renderwareai.html
- robustní třídy v C++, široký záběr
- 4-vrstvý model pro lepší návrh

AI Implant

- AI implant (http://www.ai-implant.com/)
- multiplatformní, velký záběr, 3D

Vývojové nástroje a pomůcky, kde hledat?

Existují vývojové sady (SDK – software development kit), sady nástrojů (toolkits), implementace jednotlivých algoritmů a v neposlední řadě se do popředí protlačují i skriptovací jazyky.
K mému studiu by nejlépe sloužily knihy, které se však zvláště u nás těžko shánějí a obvykle jsou drahé. U minoritních technologií je to zvykem. Věřím, že se situace změní, čemuž vývoj nasvědčuje. Seznam těch nejzajímavějších knih najdete na adrese http://www.cs.berkeley.edu/~russell/competing.html . Nejvíce mi pomohly k pochopení tutorialy (nejčastější herních vývojářů), příklady a časopisy, které popisují a ilustrují řešené problémy a principy. Pomoc lze nejlépe hledat v diskuzních fórech a newsových skupinách.