Czech Technical University in Prague

Faculty of Electrical Engineering

Department of Computer Graphics and Interaction

logocvut

Bakalářská práce

Umělá intelligence v tahových strategiích

*Lukáš Beran*

Vedoucí: titul Michal Hapala titul

Studijní program: Softwarové technologie a management

Studijní obor: Softwarové inženýrství

Leden 2011

Poděkování

First of all I would like to thank my parents, my family, and especially my little sister. She definitely gave me a different view of this world.

Additionally, I would like to thank Dr. Vlastimil Havran for his kind leadership and invaluable remarks.

Finally, I would like to express my gratitude to my few very close friends, who always tried to help me accomplish things I wanted to achieve.

Declaration – Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 17. ledna 2011 …………………………………..

Lukáš Beran

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque porta vulputate dui eget convallis. Aliquam erat volutpat. Nulla facilisi. Vestibulum ante libero, mollis ac fringilla id, malesuada vel lectus. Cras id tellus dolor, vel consectetur ipsum. Aenean dignissim, sapien et viverra fermentum, lectus enim suscipit quam, id mollis neque dolor non magna. Integer volutpat est quis erat dapibus mattis. In a purus eget ligula egestas fringilla. Aenean semper risus a dolor pretium dapibus. Donec ac justo lacus. Mauris lobortis nulla lorem, vitae porttitor sapien. Sed malesuada tempus odio vitae mollis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque porta vulputate dui eget convallis. Aliquam erat volutpat. Nulla facilisi. Vestibulum ante libero, mollis ac fringilla id, malesuada vel lectus. Cras id tellus dolor, vel consectetur ipsum. Aenean dignissim, sapien et viverra fermentum, lectus enim suscipit quam, id mollis neque dolor non magna. Integer volutpat est quis erat dapibus mattis. In a purus eget ligula egestas fringilla. Aenean semper risus a dolor pretium dapibus. Donec ac justo lacus. Mauris lobortis nulla lorem, vitae porttitor sapien. Sed malesuada tempus odio vitae mollis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Abstrakt

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque porta vulputate dui eget convallis. Aliquam erat volutpat. Nulla facilisi. Vestibulum ante libero, mollis ac fringilla id, malesuada vel lectus. Cras id tellus dolor, vel consectetur ipsum. Aenean dignissim, sapien et viverra fermentum, lectus enim suscipit quam, id mollis neque dolor non magna. Integer volutpat est quis erat dapibus mattis. In a purus eget ligula egestas fringilla. Aenean semper risus a dolor pretium dapibus. Donec ac justo lacus. Mauris lobortis nulla lorem, vitae porttitor sapien. Sed malesuada tempus odio vitae mollis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque porta vulputate dui eget convallis. Aliquam erat volutpat. Nulla facilisi. Vestibulum ante libero, mollis ac fringilla id, malesuada vel lectus. Cras id tellus dolor, vel consectetur ipsum. Aenean dignissim, sapien et viverra fermentum, lectus enim suscipit quam, id mollis neque dolor non magna. Integer volutpat est quis erat dapibus mattis. In a purus eget ligula egestas fringilla. Aenean semper risus a dolor pretium dapibus. Donec ac justo lacus. Mauris lobortis nulla lorem, vitae porttitor sapien. Sed malesuada tempus odio vitae mollis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

„Some argue that in the very long term,

rendering may best be solved by some variant

of ray tracing, in which huge numbers of rays

sample the environment for the eye’s view of each frame.

And there will also be colonies on Mars,

underwater cities, and personal jet packs. “

Real-time Rendering, Möller and Haines, 1999

ונתווקת הדבא אל דוע  
„Our hope is not yet lost”

Hatikvah, 5th verse

Obsah

[Seznam obrázků xi](#_Toc284173433)

[Seznam tabulek xi](#_Toc284173434)

[1 Úvod 13](#_Toc284173435)

[1.1 Historie 13](#_Toc284173436)

[1.2 Iluze inteligence 13](#_Toc284173437)

[2 AI algoritmy 15](#_Toc284173438)

[2.1 Stavový automat 15](#_Toc284173439)

[2.2 Goal – driven 15](#_Toc284173440)

[2.3 Fuzzy logika 15](#_Toc284173441)

[2.4 Umělý život 15](#_Toc284173442)

[2.4.1 Mazlíčci 15](#_Toc284173443)

[2.4.2 Společenské simulace 15](#_Toc284173444)

[2.4.3 Hry na Boha 16](#_Toc284173445)

[2.4.4 Evoluční hry 16](#_Toc284173446)

[2.4.5 A-Life a tahové strategie 16](#_Toc284173447)

[2.5 Genetické algoritmy 16](#_Toc284173448)

[2.6 Neuronové sítě 16](#_Toc284173449)

[2.7 Rozhodovací stromy 16](#_Toc284173450)

[2.8 Hledání cest 16](#_Toc284173451)

[2.8.1 Prohledávání do šířky 17](#_Toc284173452)

[2.8.2 Pár slov k A\* 18](#_Toc284173453)

[2.9 Skriptování 18](#_Toc284173454)

[2.9.1 Co je to skript? 18](#_Toc284173455)

[3 Frameworky 18](#_Toc284173456)

[4 Implementace 18](#_Toc284173457)

[4.1 Volba prostředí 18](#_Toc284173458)

[5 Pozorování 18](#_Toc284173459)

[6 Závěr 18](#_Toc284173460)

[7 Citovaná literatura 19](#_Toc284173461)

[A Ukázky kódu 19](#_Toc284173462)

[B Obsah CD 21](#_Toc284173463)

Seznam obrázků

Nenalezena položka seznamu obrázků.

Seznam tabulek

Nenalezena položka seznamu obrázků.

Seznam grafů

Nenalezena položka seznamu obrázků.  
Seznam ukázek kódu

Nenalezena položka seznamu obrázků.

# Úvod

V první kapitole, v druhé kapitole…

## Historie

Vývoj umělé inteligence v počítačových hrách započal s hrami samotnými. Již první grafická hra Tic Tac Toe, piškvorky 3 x 3 pole, z roku 1952 měla v sobě zabudovanou umělou inteligenci protihráče. (1) Totéž platí i pro následovníky „Tenisu pro dva“ nyní známějším pod jménem Pong, nebo např. pro hru Pac Man.

Přestože v počátcích video her byly hry značně jednoduché včetně umělé inteligence, mnohé by mohla překvapit propracovanost a složitost umělé inteligence ve hře Pac Man. Jedná se o jednoduchou hru odehrávající se na jedné obrazovce, kde máte za úkol sbírat po bludišti kolečka představující jídlo a zároveň se nenechat chytit jedním ze 4 duchů.

Duchové nejen mají různé barvy, jména a přezdívky, ale také odlišné chování, čehož si málokdo z hráčů na první pohled všimne. (2)

Složitější umělé inteligence bylo potřeba vyvíjet právě s rozvojem strategických her (Civilization, Heroes of Might and Magic), kde bylo a je nutností udělat poměrně složitou AI, aby hra byla vůbec hratelná.

V minulosti byl vývoj umělé inteligence v ústraní. Mnohem důležitější bylo vyvíjet grafickou stránku hry, která zároveň znatelně vytěžoval CPU počítačů a tedy ani výpočetní výkon nezbýval pro AI. Vývoj umělé inteligence často probíhá až v posledních pár měsících tvorby hry, a tedy proto nemůže být AI dokonalá.

V současné době, kdy už je hráč nasycen téměř dokonalou fotorealistickou grafikou, kdy výkon počítačů je mnohem dál, je již dostatek prostoru pro rozvoj AI v počítačových hrách. (3)

## Iluze inteligence

Umělá inteligence v počítačových hrách má s umělou inteligencí všeobecně mnoho společného. Metody jako rozhodovací stromy, konečné stavové automaty nebo třeba neuronové sítě můžete najít v obojím.

AI ve hrách se v určitých aspektech liší od umělé inteligence např. řízení leteckého provozu. Nemusí být co nejchytřejší, co nejlepší. Již dávno není problém udělat inteligenci bota ve FPS střílečce neomylného, který by vždy dokázal při spatření hráče ho zabít jednou ranou z pistole, head shotem. Podobně lze udělat nepřekonatelného střelce v basketbalu, který trefí koš přes celé hřiště. Oba dva případy by potencionální hráče brzy odradily.

Existují hráči, kteří až absurdní obtížnost ocení, ale většina hráčů upřednostní, když polovinu času vyhrávají a polovinu času prohrávají. Vůbec nemusí být špatným nápadem korigovat obtížnost hry dle počtu výher a proher a tím jejich poměr zachovávat přibližně roven jedna ku jedné.

Je důležité zachovat určitou reálnost chování NPC, hráč by neměl mít pocit toho, že soupeř podvádí. Při vývoji hry Empire Earth měli testeři hlásit jakékoliv podezřelé chování soupeře, měli zapsat kdykoliv si mysleli, že jejich soupeř podvádí. Přestože se vždy nejednalo o podvod, tak se nahlášené problémy vždy prodiskutovali a případně se umělá inteligence poté upravila. (4)

Podvádění soupeřů někdy nemusí být pro škodu, ale nesmí to soupeř poznat. Co to vlastně je to podvádění ve hrách? NPC z pohledu vývojaře podvádí, pokud využívá věcí ve hře, které nemůže využít lidský hráč. Např. když od počátku hry soupeř ví, kde má hráč základnu a vysílá tam bez předchozího průzkumu své vojáky. Případně ve strategických hrách není neobvyklým jevem, když soupeř vytvoří pro obranu své základny jednotky z ničeho, ze surovin, které nemohl během hry získat. Oba ze zmíněných podvodů mohou být přípustné a zlepšit celkovou hratelnost hry, pokud je hráč nezaznamená.

Z pohledu laika může být podvádění i to, když se soupeř chová až moc dokonale, nejedná realisticky. Sem patří již zmíněná dokonalá střelba v FPS. Realističnost je z pohledu hráče hodně důležitá, ale jsou případy, kdy by mu mohla přijít otravná. Adventura, kde jsou dvě cesty na konci každé z nich je truhla, v jedné z nich je klíč od té druhé. Hráč může zvolit jakoukoli z těchto dvou cest jako první. Pokud by zvolil jako první tu, jež ho vede k zamčené truhlici, musel by se vracet, jít druhou cestou, získat klíč a opět se vracet k první truhle. Tento nedostatek lze elegantně vyřešit tak, že první truhla, kterou hráč otevře bude obsahovat klíč od té druhé truhly. (5)

K dalším zajímavým přístupům patří podívat se na řešení problému AI z jiného pohledu. Nesnažit se dělat NPC přehnaně komplexní a inteligentní, ale vložit tu inteligenci do světa kolem ní. Známým příkladem je hra The Sims, simulátor lidí, kteří mají své potřeby jako je hlad, jež je potřeba uspokojit. Pokud bychom se na to podívali z pohledu reálného světa, tak postava, když dostane hlad, tak najde v domě nejbližší jídlo (lednička, hotové jídlo na stole), dojde k němu a začne jíst. V The Sims je to dělané jinak. Lednička do určitého poloměru vysílá zprávu „Můžu uspokojit tvůj hlad“. Pokud se simík dostane blízkosti ledničky, vyhodnotí své aktuální potřeba a jestli mu v jeho blízkosti něco oznamuje, že je může uspokojit, pokud ano, tak se přesune k nim. (5)

Do prostředí lze zanést mnohé rozmanité informace. Např. bodliny v 2D plošinovce, které ubírají životy všemu, co se jich dotkne. Místo programování logiky do hráče, že když je v blízkosti bodlin, změň animaci hráče, uber životy, případně uskoč se tato logika přiřadí do logiky bodlin. Tedy bodliny vědí, že mají hráči ubrat životy, změnit mu animaci, odstrčit ho. Tento přístup nejen zjednodušuje komplexnost logiky hráče, ale zjednodušuje přidávání nových herních prvků do hry.

Co je důležité si zapamatovat pro tvorbu herní AI je to, že hra nemusí být dokonale realistická, férová, neporazitelná. Pořád se pohybujeme v herním, v zábavném průmyslu a tedy dobrá AI má za úkol hlavně pobavit a být výzvou pro hráče. Musí umět bavit naprostého nováčka, i zkušeného hráče hrající online turnaje.

# AI algoritmy

## Stavový automat

## Goal – driven

## Fuzzy logika

## Umělý život

Artificial Life, A life, umělý život. Další ze způsobů, který lze využít pro tvorbu AI, ale i jako základní princip hratelnosti. Jedná se o simulaci života, kde hráč přímo, či nepřímo ovládá jednu, nebo několik životních forem(lidí, zvířat, příšer), které bez zásahu hráče dokážou ve hře samostatně myslet, pohybovat se, žít.

Prapůvod umělého života je v Conveyově simulaci Game of Life. Jde o svět tvořený 2D mřížkou. Každá buňka mřížky představuje buď živou, nebo mrtvou buňku. Život se v tomto světě rozvíjí v tazích, kolech, kde několik pravidel určuje, jestli v následujícím tahu vznikne nová buňka na místě, kde žádná není, nebo zanikne tam, kde je, nebo jen stávající buňka bude pokračovat ve svém životě. Pravidla pro život v dalším kole jsou celkem čtyři a odvíjí se pouze podle počtu živých buněk v okolí v kole současném.

Přestože se jedná o jednoduchou simulaci, bylo v tomto světě o 4 pravidlech objeveno několik stovek vzorů skupin buněk se specifickým chováním rozdělených do několika kategorií. Patří jsem např. glider, skupina buněk, která vlivem zanikání a vzniku nových buněk působí dojmem, že se celá skupina pohybuje jedním směrem. (4) Game of life není stále zcela prozkoumána, každým rokem vědci z celého světa objevují nové vzory.

Simulaci života můžeme vidět v mnohých hrách, které jdou rozdělit do několika skupin. (5)

### Mazlíčci

Žánr, kde se staráte o své virtuální zvířátko, reálné, či smyšlené. Musíte ho krmit, cvičit, za prohřešky trestat. Zvíře je schopné se od hráče učit. Mazlíčci jsou většinou roztomilí a svými emocemi, chováním dávají hráči najevo, co s nimi má dělat.

Oproti jiným A-Life hrám se zvířátka často nemůžou rozmnožovat, umírat. Neznamená to ale, že nemůžete v takových hrách dělat špatná rozhodnutí. Pokud se chováte špatně ke svěřenému zvířeti, může od vás utéct.

Mezi známé zástupce tohoto žánru patří např. Tamagotchi, Nintendogs, nebo Neopets.

### Společenské simulace

Kdo by neznal hru The Sims, která odstartovala tento žánr. Úkolem je starat se o jednu, čí více osob, které mají několik základních potřeb (hygiena, hlad, zábava, potřeba wc, kamarádi apod.), jež je třeba udržovat. Na rozdíl od starání se o mazlíčky v těchto hrách jsou důležité sociální vazby s ostatními osobami v rodině a v sousedství, které je třeba udržovat a rozvíjet pro lepší pokrok ve hře.

### Hry na Boha

Populous, Black and White, Dungeon Keeper, zástupci tohoto žánru. V Dungeon Keeperu se impové sami od sebe se starají o zpevňování stěn, těžbu drahokamů, či zabírají nová území. Vy jako „Bůh“, je ale můžete trestat, obětovávat, dávat jim nepřímo rozkazy, nebo je jen sledovat. I ostatní příšerky žijí svým životem, chodí spát, jíst, cvičit se a nechávají si od vás zaplatit za obranu vašeho království. Vy je můžete vzít a přesunout tam, kde jsou zrovna potřeba. Např. poslat je objevovat nová kouzla, nebo zdokonalovat se v útoku.

### Evoluční hry

Hry, v nichž se staráte o populace několika generací, které se množí vylepšují, přenášejí nové lepší vlastnosti do dalších generací. V těchto hrách můžete nechat organismy se vyvíjet vlastní cestou, nebo můžete do vývoje zasahovat změnou prostředí, či přidáním nově navržených vlastních forem života.

Populárními hrami jsou např. Spore, či Evolution : The Game of Intelligent Life .

### A-Life a tahové strategie

Původní experiment Game of Life se odehrává v jednotlivých krocích, ale stěží to lze označit za hru, jde jen o matematický experiment.

Nepodařilo se mi nalézt žádnou tahovou strategii spojovanou s metodou umělého života. Důvodem může být, že ve všech zmíněných hrách bylo podstatné, aby se bez hráčova zásahu vše hezky hýbalo, žilo vlastním životem a aby to bylo zábavné jen sledovat. Sledovat v reálném čase.

A to může být hlavní překážkou pro zdárné využití této myšlenky v tahových strategiích, kde čekání na odkliknutí dalšího tahu by narušovalo ten plynulý život.

## Genetické algoritmy

## Neuronové sítě

## Rozhodovací stromy

## Hledání cest

Jedná se o speciální problém, který spadá do problematiky umělé inteligence. Hráč chce vyslat jednotky do soupeřovi základny. Neurčuje jednotkám jednotlivé kroky, určí jen požadovanou pozici a zbytek se udělá automaticky.

Úkol je zřejmý, nalézt nejoptimálnější a zároveň věrohodnou cestu mezi dvěma místy na mapě. Nejoptimálnější často znamená nejkratší cestu, ale nemusí tomu být tak. Např. pokud v nejkratší cestě bude minové pole, tak by se mu měli jednotky umět vyhnout. Minimálně jednotky ovládané počítačem.

Zachování věrohodnosti chování a přirozenosti je také důležité. Nemusí působit dobře, pokud jednotky se pohybují s naprostou přesností k cíli a obcházejí horu, kterou ještě hráč prozatím neobjevil, a tedy jednotky by o ní neměli vědět.

K dalšímu problému, který je třeba v této kategorii řešit, je přesouvání více jednotek naráz. Působilo by nepřirozeně, kdyby všechny jednotky se snažili jít nejkratší možnou cestou, a tedy šli jeden za druhým jako vláček.

S hledáním cest je nemálo problémů, ale pro řešení základní úlohy, hledání nejkratších cest, bylo již vymyšleno několik funkčních algoritmů.

Hledání (nejkratších) cest je jedním ze stěžejních problémů teorie grafů. V tomto případě grafem se nemíní grafické zobrazení tabulkových hodnost na osách x a y, ale graf je množina uzlů a hran mezi nimi. Příkladem grafu může být bitevní pole složené ze šestiúhelníků z tahové strategie HOMAM 3, kde co jeden šestiúhelník, to jeden uzel grafu. Každý uzel má až šest hran, které znázorňují možnost pohybu na sousední šestiúhelníky.

Mezi algoritmy řešící tuto úlohu patří prohledávání do šířky, prohledávání do hloubky, heuristické prohledávání(sem patří ve hrách velice populární algoritmus A\*), Dijkstrův algoritmus, Floyd-Warshallův algoritmus, Bellman-Fordùv algoritmus.

### Prohledávání do šířky

Prohledávání do šířky je poměrně jednoduchý algoritmus snadno použitelný v počítačových hrách. Průběh vyhledávání připomíná vlnu rozšiřující se na hladině vody, když na její povrch dopadne kapka vody.

Základní princip algoritmu vysvětlím na příkladě čtvercové 2D mapy, kde je označen jeden čtverec jako začátek(např. by to mohlo být aktuální pozice jednotky), jeden čtverec jako cíl(např. místo, kam hráč klikl myší a chce poslat jednotku) a některé ze čtverců jsou označené jako neprůchozí zdi. V naší imaginární hře s 2D mapou se můžou jednotky pohybovat pouze do 4 směrů, ne diagonálně.

Vyhledávání probíhá v několika krocích. Prvně se zkontroluje, jestli se začátek neshoduje s cílem, pokud ano, „cesta“ nalezena. Pokud ne, podívám se postupně na jednotlivé sousedy startovního pole, jestli ony nejsou cílem. Až zkontroluji všechny a nenacházím-li mezi nimi cíl, tak zkontroluji sousedy sousedů startovního pole. Když ani poté cíl nenacházím pokračuji stejným způsobem, otestuji všechny čtverce vzdálené na tři kroky od startu. Tento postup se opakuje dokud se nenarazí na cíl, nebo se nezjistí, že cíl není dosažitelný(např. je na ostrově, na který nevede most).

Jak se pozná, že k cíli se ze startu nelze dostat? Při procházení mapy „vlnou“ od startu si označuji čtverce, na kterých už jsem byl a hledal cíl, pokud se dostanu do fáze, že všechna pole dostupná ze startu jsou označená a ani jedno z nich nebylo cíl, značí to, že cíl je nedostupný.

Tento algoritmus s jistotou najde nejkratší cestu do cíle, pokud existuje a umí i zjistit, že taková cesta neexistuje. Jeho nevýhodou je, že hledá cíl ve všech směrech a prohledává velké množství prostoru, což se projeví především ve velkých prázdných plochách bez zdí. Pokud start a cíl budou vzdálené deset polí, tak než se cíl nalezne, projdou se všechna pole do vzdálenosti devět od startu. Druhý problém je ještě výraznější, pokud na mapě velké tisíc krát tisíc polí neexistuje cesta mezi startem a cílem, tak se prohledá až milión polí než se zjistí, že cesta neexistuje.

První problém lze částečně vyřešit vysláním vln zároveň z cíle a startu proti sobě. U příkladu polí vzdálených deset polí od sebe můžeme porovnat obsahy dvou kruhů o poloměru pět a jednoho o poloměru deset. Konstantu pí můžeme vynechat a máme to 2 \* 5 \* 5 = 50 ku 10 \* 10 = 100, tedy u dvou vln se v tomto případě prohledá polovina polí oproti vlně jedné.

Zamezit prohledávání miliónu polí lze jednoduchou podmínkou. Pokud se nenalezne cíl do vzdálenosti 30, 50, … vyhledávání se ukončí jako neúspěšné. Toto vylepšení není ideální, protože teď algoritmus nezaručuje, že cestu nalezne, přestože existuje. Pokud zvolíte limit 50, co když by byl cíl ve vzdálenosti 51? Hranici, kdy by mělo vyhledávání skončit, není jednoduché najít.

### Pár slov k A\*

A\* patří k nejpoužívanějším vyhledávacím algoritmům v počítačových hrách. S algoritmem prohledávání do šířky má více společného než se na první pohled může zdát. Prohledávání do šířky je vlastně speciálním případem A\*.

U prohledávání do šířky jsme postupně kontrolovali nejdříve všechny čtverce vzdálené jedna od startu, poté 2, 3, 4, atd. V jakém pořadí se budou kontrolovat čtverce, bylo dáno pouze jejich vzdáleností od startu. U algoritmu A\* k tomu přibude druhý parametr, a to předpokládaná vzdálenost k cíli. Čtverce se prohledávají v pořadí dané součtem vzdálenosti od počátku a předpokládané vzdálenosti do cíle. Čtverec, který má tento součet menší než jiný bude testován před ním.

Pod předpokládanou vzdáleností si lze představit odhad vzdálenosti se zanedbáním terénů, zdí mezi kontrolovaným čtvercem a cílem. Např. u příkladu se čtvercovou mapou dobře funguje jako odhad tzv. Manhattonská metoda. Odhad touto metodou se spočítá jako součet absolutních hodnot rozdílu x-vé a y-vé souřadnice zkoumaného čtverce a cíle. Příklad: čtverce se souřadnicemi [3, 4] a [0, 7] mají vzdálenost |3-0| + |4-7| = 6. Mezi další metody patří např. euklidovská vzdálenost.

Algoritmus lze snadno vylepšit, aby podporoval různé typy povrchů (cesta , bažina), aby znemožňoval dokonalou navigaci v neprozkoumaném prostředí apod.

## Skriptování

### Co je to skript?

# Frameworky

# Implementace

## Volba prostředí

# Pozorování

# Závěr

# Citovaná literatura

1. **Lackore, Jason.** Survey on the use of artificial intelligence in video games. *Jason's Website.* [Online] [Cited: leden 20, 2011.] http://jlackore.com/Documents/Survey%20on%20the%20Use%20of%20Artificial%20Intelligence%20in%20Video%20Games.pdf.

2. **Birch, Chad.** Understanding Pac-Man Ghost Behavior. *Game Internals.* [Online] prosinec 2, 2010. [Cited: leden 20, 2010.] http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior.

3. **Tozour, Paul.** The Evolution of Game AI. [book auth.] Steve Rabin. *AI Game Wisdom.*

4. **Scott, Bob.** The Illusion of Intelligence. [book auth.] Steve Rabin. *AI Game Wisdom.*

5. **Kirby, Neil.** Solving the right problem. [book auth.] Steve Rabin. *AI Game Wisdom.*

6. Glider. *Conway Life.* [Online] červenec 23, 2010. [Cited: leden 20, 2011.] http://www.conwaylife.com/wiki/index.php?title=Glider.

7. Life Simulation Game. *Wikipedia.* [Online] leden 17, 2011. [Cited: leden 20, 2011.] http://en.wikipedia.org/wiki/Life\_simulation\_game.

8. **Belis, Mary.** Computer and Video Game History. *About.com Inventors.* [Online] [Cited: leden 20, 2011.] http://inventors.about.com/library/inventors/blcomputer\_videogames.htm.

1. Ukázky kódu
2. Obsah CD

|  |  |
| --- | --- |
| ┐ ├───results  │ ├──cell  │ └──x86  ├───src  │ ├──cell rt  │ ├──x86 client  │ └──x86 rt  └───thesis | Data files with results measured on Cell  Data files with results measured on x86  IBM Cell Ray Tracer source  x86 to Cell Client source  x86 Ray Tracer source  PDF and Microsoft Word versions of this text |