

DARK OBERON

Projektová dokumentácia

Valéria Šventová
Martin Košalko

Obsah

1.1	ÚVOD	1
1.2	STRUČNÝ POPIS PROJEKTU	1
1.3	VÝVOJOVÝ TÍM.....	1
1.4	VÝVOJ	2
1.4.1	<i>Vývojové prostriedky</i>	2
1.4.2	<i>Chronológia vývoja</i>	3
1.4.3	<i>Pôvodné zámary projektu a výsledok – porovnanie</i>	4
1.4.3.1	Konkrétna hra verzus výpočtový stoj na hry	4
1.4.3.2	Inteligentný počítačový hráč	4
1.4.3.3	Grafika	5
1.4.3.4	Prenositel'nosť	6
1.4.3.5	Sieťovanie.....	6

1.1 Úvod

Projekt Dark Oberon bol vypracovaný v rámci predmetu „PRG023 – Projekt“ na MFF UK pod vedením RNDr. Jakuba Yaghoba.

1.2 Stručný popis projektu

Cieľom projektu Dark Oberon bolo naprogramovať strategickú hru, ktorá sa svojim charakterom podobá hrám, ako sú napríklad Warcraft, či C&C RedAlert. Každý hráč môže, ako je to v takýchto hrách bežné, vyrábať rôzne typy jednotiek a budov, vysielat' formácie svojich jednotiek proti nepriateľovi a pod. Víťazom sa stáva hráč, ktorý zničí všetky jednotky protihráča.

Hra Dark Oberon má len multiplayer mód s komunikáciou cez lokálnu sieť. V rámci hry bol tiež implementovaný experimentálny počítačový hráč.

Pre grafický výstup bolo použité rozhranie OpenGL.

DarkOberon poskytuje užívateľovi široké možnosti tvorby úplne novej hry tohoto typu a to prostredníctvom konfiguračných súborov, v ktorých sa dajú nadefinovať mnohé vlastnosti jednotiek, či samotnej hry.

Celá hra je prenositeľná na platformy Windows, Unix (X11).

1.3 Vývojový tím

Už od začiatku bol projektový tím tvorený šiestimi členmi. Tento počet sa počas vývoja nezmenil. Nasleduje prehľad členov tímu s popisom ich práce na projekte:



Marián Černý (jojo@matfyz.cz)

Návrh a tvorba sieťovej komunikácie, zabezpečenie prenositeľnosti na systém Unix, správa CVS, projektové web stránky, fotografovanie modelov, dokumentácia.



Peter Knut (peter.knut@matfyz.cz)

Návrh a implementácia grafického rozhrania hry, Data a Map editory, design web stránok, modely rasy Human, spracovanie textúr, návrh a implementácia dátových súborov, návrh konfiguračných súborov, dokumentácia.

**Martin Kořalko** (cauchy@matfyz.cz)

Návrh a implementácia systému správ jednotlivým jednotkám hry, fronta správ, návrh interpretácie výsledkov neurónových sietí, modely schémy Plastic, Map editor, implementácia konfiguračných súborov, dokumentácia.

**Michal Král** (index@matfyz.cz)

Návrh a implementácia inteligentného počítačového hráča, tvorba neurónových sietí, dokumentácia.

**Jiří Krejsa** (crazy@matfyz.cz)

Návrh a implementácia algoritmov strieľania a chodenia, bazén vlákien, dokumentácia.

**Valéria Šventová** (liberty@matfyz.cz)

Návrh a implementácia algoritmov pre akcie jednotiek, testovanie, dokumentácia.

Rozdelenie činností spočiatku nebolo presne stanovené, boli rozdelené len hlavné smery vývoja (grafika a užívateľský interface (1), sieťová podpora a prenositeľnosť (1), algoritmy týkajúce sa priamo samotnej hry (2-3), počítačový hráč (1-2)). Postupom času sa objavovali nové úlohy a tie už existujúce sa presnejšie kryštalizovali.

1.4 Vývoj

1.4.1 Vývojové prostriedky

Ako programovací jazyk sme si zvolili jazyk C++ a jeho objektových vlastností sme v projekte bohaty využili. Grafický výstup je implementovaný použitím rozhrania OpenGL (www.opengl.org). Prenositeľnosť aplikácie zabezpečuje knižnica GLFW

(glfw.sourceforge.net). Pre zvukový výstup je použitá knižnica FMOD (www.fmod.cz). Zdrojové súbory projektu sú uložené na CVS servri služby SourceForge.NET (www.sourceforge.net).

1.4.2 Chronológia vývoja

Chronologický popis priebehu prác je približný, mnohé zmienené činnosti presahovali uvedený časový rozsah. Spočiatku bolo zamýšľané v rámci softwarového projektu vytvoriť len jedinú hru, neskôr sme prešli na tvorbu univerzálneho engine pre hry. Požiadavky boli maximálne: všeobecnosť, čo najmenšia obmedzenosť a konfigurovateľnosť. Postupom času sa však ukázalo, že príliš veľká všeobecnosť nesie so sebou nemalé problémy, napríklad náročnú implementáciu mnohých algoritmov. Preto bolo od týchto požiadaviek často upustené. V nasledujúcom popise vývoja projektu DarkOberon sú obsiahnuté aj veľké zmeny, ktorými projekt prešiel.

- **10. október 2002** – prvá informačná schôdzka členov projektu (z ktorej vznikla myšlienka vytvoriť strategickú hru), voľba jazyka, prvé rozdelenie úloh,
- **koniec októbra 2002** – oficiálne vypísanie projektu komisiou,
- **november 2002 – január 2003** – návrh vnútornej koncepcie, rozhodovanie sa nad jednotlivými krokmi implementácie základných častí hry (mapy, počítačoví hráči, správa jednotiek hráčov, vlastnosti jednotlivých akcií jednotiek (chodenie, strelba, tvorba nových jednotiek, ťaženie materiálov a pod.)), registrácia projektu na SourceForge a začiatky používania CVS, vznik prvých súborov,
- **február – jún 2003** – implementácia vnútorných štruktúr hry, algoritmu chodenia, základných algoritmov pre jednotlivé činnosti jednotiek, grafického rozhrania hry. Projekt prekonáva prvú výraznejšiu zmenu: pôvodne bolo zamýšľané vytvoriť viacero segmentov (úrovní, v ktorých sa môžu jednotky pohybovať), ich počet by bol zadávaný prostredníctvom konfiguračných súborov. Z dôvodu príliš obtiažnej implementácie sa prešlo k naimplementovaniu presne troch segmentov predstavujúcich podzemie, povrch zeme a vzduch. Navyše sa predpokladá, že najvyšší segment (vzduch) obsahuje polopriehľadné alebo priehľadné textúry, zvyšné segmenty môžu používať ľubovoľné textúry,
- **október 2003 – január 2004** – na začiatku tohto obdobia projekt prekonal ďalšiu zmenu: myšlienka všeobecných závislostí ustúpila do úzadia a nahradila ju závislosť na predkoch a množstve materiálov. U všeobecnej závislosti sa predpokladá, že vlastnosti jednotky (budovy) sú závislé na existencii iných jednotiek (budov). Závislosť na predkoch očakáva jedine existenciu samotnej jednotky, na ktorú bude upgrade postavený. Toto obdobie charakterizujú tiež počiatky návrhu inteligentného počítačového hráča. Uskutočnila sa schôdzka s Mgr. Romanom Nerudom, kde sa rozoberali možnosti a úskalia implementácie inteligentného počítačového hráča využitím viacvrstvových perceptronových sietí. Dokončovali sa tiež základné algoritmy pre jednotlivé jednotky.
- **február – jún 2004** – po dôkladnej analýze sme zmenili architektúru hry: od Update funkcií sa prešlo frontu správ. Update funkcia bola volaná pre každú jednotku v cykle až do ukončenia hry a prebiehali v nej všetky akcie spojené s touto jednotkou. Fronta správ má iný charakter – vkladajú sa do nej správy od všetkých jednotiek, každá správa má časovú značku a predstavuje nejakú akciu jednotky. Zároveň obsahuje informáciu, pre koho je určená. Správy sú z fronty vyberané podľa časovej značky a

sú spracovávané. Bližší popis fungovania fronty správ je popísaný v samostatnej dokumentácii.

Obdobie od apríla charakterizuje implementácia neurónových sietí (počítačový hráč), obnova grafického rozhrania – nové textúry vznikali fotením modelov jednotiek, ktoré boli vyrobené z plastelíny; testovanie už naprogramovaných častí,

- **október – december 2004** – zavedenie viacvláknovosti (multithreading, v „hlavnom“ vlákne beží užívateľský vstup a grafický výstup, ostatné vlákna spracovávajú napríklad správy, chodenie, počítanie vzdialeností a pod.), štúdium problematiky sietí, návrh a implementácia, testovanie naprogramovaných neurónových sietí a prvé pokusy o interpretáciu ich výsledkov,
- **január – máj 2005** – testovanie, dokumentácia, programovanie sieťovej podpory pretrváva, doladovanie detailov.

Počas celej doby trvania projektu prebiehali takmer pravidelné projektové schôdzky, s výnimkou letných prázdnin. Frekvencia stretnutí bola ovplyvňovaná intenzitou vývoja projektu. V druhom roku vývoja schôdzky prebiehali pravidelne jedenkrát týždenne vo večerných hodinách. Zároveň jedenkrát za dva týždne prebehla schôdzka s vedúcim projektu. Ku koncu projektu sa intenzita posledne spomínaných schôdzok zvýšila na raz týždenne.

1.4.3 Pôvodné zámery projektu a výsledok – porovnanie

1.4.3.1 Konkrétna hra verzus výpočtový stroj na hry

Pôvodným zámerom vývojového tímu bolo vytvoriť strategickú hru typu Warcraft 2, ktorá by bola modifikovateľná vstupnými dátami. Podľa špecifikácie malo byť umožnené meniť vlastnosti konkrétnych jednotiek (vdzucholod' mohla lietať rýchlejšie, bojovník mal dostať väčšiu silu...). Ukázalo sa, že pre užívateľa môže byť obmedzujúce mať iba obmedzený preddefinovaný počet typov jednotiek, a preto sme umožnili definovať ľubovoľný počet typov jednotiek s užívateľsky definovanými vlastnosťami. To však kládlo vyššie nároky na programovanie, pretože o typoch jednotiek, ktoré si užívateľ definuje nie je možné predpokladať takmer nič. Sklízli sme teda k programovaniu všeobecného stroja na strategické hry. Niektoré algoritmy sme však nedokázali (prípadne to nebolo žiadúce a efektné) zovšeobecniť úplne, a tak sme z požiadaviek ubrali. Príkladom algoritmu, ktorý nebolo vhodné implementovať všeobecne je grafické zobrazenie neobmedzeného počtu segmentov (sklízlo sa teda k pevnému počtu segmentov). Výsledkom je teda takmer všeobecná šablóna – výpočtový stroj na strategické hry bežiacie v reálnom čase (takzvané RTS).

1.4.3.2 Inteligentný počítačový hráč

V počiatkoch vývoja projektu sme chceli implementovať inteligentného počítačového hráča, ktorý by bol schopný „odpozorovať“ stratégiu hry od svojich protihráčov. Výsledky každej hry sa mali ukladať, následne spracovať a vyhodnocovať. Podľa dosiahnutých výsledkov sa potom mali modifikovať neurónové siete rozhodujúce o stratégii počítačového hráča a tým sa mal zabezpečiť jeho rozvoj.

Ako prvý problém sa ukázalo vyhodnotenie odohranej hry, ktoré je celkom netriviálne. Je totiž problém rozhodnúť, či daná akcia vykonaná hráčom (prípadne neurónovou sieťou) v nejaký okamih hry bola vo výsledku pozitívna a nájsť (a vyčísliť) mieru pozitívnosti danej akcie. Od idey ukladania hry a jej vyhodnocovania sa teda upustilo a nie je implementované.

Už na schôdzke s Mgr. Romanom Nerudom sme boli upozornení na to, že siete, ktoré prichádzali do úvahy pre rozhodovanie o činnosti hráča môžu byť vzhľadom na vstupné

parametre (ich počet) a počet naučených príkladov netriviálne mohutné. Z netriviálnej mohutnosti sietí okamžite vyplýva aj nezanedbateľná dĺžka učenia sietí. V praxi sa ukázalo, že na to, aby sme sieť dokázali naučiť všetkým požadovaným prípadom je ozaj nutná obrovská sieť (čo ešte nebol až taký veľký problém) a veľký počet vzorových príkladov, ktoré sa vzájomne prelínajú. Najväčší problém bolo nájsť ohodnotenie vzorových príkladov (potrebných pre algoritmus spätnej propagácie – back propagation) tak, aby sa ich sieť bola schopná naučiť.

Výsledok nášho snaženia je teda implementácia alebo skôr pokus o implementáciu inteligentného počítačového hráča pomocou systému viacvrstvových počítačových sietí. Niektoré akcie hráča sa nám podarilo zvládnuť lepšie (ťaženie materiálov, stavanie bojových jednotiek a potrebných budov), iné menej (útočenie). Naše pôvodné predstavy boli však vysoko nadhodnotené a neboli sme ich schopní úplne naplniť. Implementácia inteligencie je svojou zložitou a komplexnosťou pravdepodobne hodná samotného softwarového projektu, ktorý by sa dal založiť na už existujúcej hre.

1.4.3.3 Grafika

Grafická stránka projektu naplnila naše pôvodné zámery. Rozhranie OpenGL spoľahlivo fungovalo na všetkých testovaných systémoch. V súlade so špecifikáciou projekt podporuje viacvrstvomú grafiku. V plnej kráse je to viditeľné v „multiview“ pohľade na mapu, kde sú jednotky zo spodného segmentu zobrazované polopriehľadne (využitie alfa-kanálu).

Všetky elementy prostredia a jednotky hráčov sú zobrazované ako 2D obrázky – textúry. Pre vytvorenie ukážkovej hry sme teda potrebovali veľké množstvo textúr (obrázky budov, panáčikov, stromov). Prvým nápadom bolo vytváranie 3D modelov v programe 3DStudio Max a ich následná konverzia do potrebného formátu. To sa ukázalo ako neefektívne a časovo náročné. Hľadali sme teda časovo menej náročný spôsob, ktorým mohlo byť fotografovanie reálnych modelov. Prešli sme mnohým nápadom (lego, hračky „igraček“, reálne ľudské postavy...) až prišiel nápad vymodelovať celé prostredie z plastelíny. Tento nápad sme (aj vďaka externým „členom projektu“ a trpezlivej práci grafika) v plnej miere zrealizovali. Modely bolo nutné vymodelovať (čo bolo príjemným sprestrením práce), vyfotiť ich zo všetkých strán pri každej činnosti a následne fotografie spracovať do požadovaného formátu. Výsledkom je teda plastický svet, ktorý je originálny a oku lahodiaci.



Obrázok 1: Počas fotenia prasiatka

Vzhľadom na veľké množstvo textúr, ktoré je potrebné pre animácie všetkých činností jednotiek, nie sú všetky jednotky úplne animované. Snažili sme sa však, aby bolo možné každú akciu ukázať aspoň na jednej jednotke.

1.4.3.4 Prenositeľnosť

V špecifikácii projektu sme uviedli, že program by mal byť prenositeľný na platformy Windows a Unix/X11. Zdalo sa nám totiž, že tento typ programov na unixových systémoch chýba. Cieľ sa nám podarilo v plnej miere splniť vďaka použitým knižniciam, ktoré zabezpečujú vstupy a výstupy (GLFW, FMOD) a aj vďaka používaniu štandardných C/C++ funkcií.

Program boli vyvíjané na platforme Win32 (v MS Visual Studio 6.0, neskôr MS Visual Studio.NET 2003), na Unixových platformách FreeBSD a Mandrake Linux (vim, gcc). Na týchto platformách program funguje (FreeBSD bez zvuku). Program by mal byť prenositeľný aj na všetky platformy uvedené v špecifikácii GLFW a FMOD.

1.4.3.5 Sieťovanie

Pri tvorbe špecifikácie projektu sme sa zhodli len na tom, že hra by mala mať multiplayer mód a mala by bežať po lokálnej sieti. Tento zámer sa podarilo naplniť s čím môžeme byť spokojní.

V priebehu vývoja projektu však vzniklo niekoľko výborných nápadov týkajúcich sa sietí, no nepodarilo sa ich implementovať. Chceli sme počítače v sieti (dynamicky) prepájať tak, aby bolo zaťaženie počítačov a liniek optimálne. S tým súvisí aj umiestnenie počítačových hráčov, ktorí mali byť distribuovaní medzi všetky počítače podľa výpočtového výkonu – v súčasnosti bežia všetci na počítači tvorca hry. Podobne existoval zámer, že po odpojení sa niektorého z hráčov jeho funkciu preberie počítačový hráč bežiaci na inom (ešte pripojenom) počítači – v súčasnosti sa po odpojení hráča všetkým vzdialeným počítačom zašle správa o odpojení, čím sa hráč všade deaktivuje (jeho jednotky korektne umrú).

Je možné povedať, že v prípade dostatku času by sa dalo mnoho vecí vylepšiť.