

Predmet:	PGP		
Autor:	Radakovič Jozef	xradak00	
		Dátum vypracovania:	15. január 2008
Projekt:	počítačová hra "Balls"		
Cieľ:	Hlavný cieľ je úspešne nadviazať na projekt z predmetu PGR a MUL. Hlavné zameranie je vylepšenie optimalizácie zobrazovania a pridanie niektorých efektov.		
Interný názov:	Balls		
www	http://eva.fit.vutbr.cz/~xradak00/balls		

1.	Predstavenie projektu	2
2.	Zadanie	2
3.	Riešenie	2
3.1.	Efekt slnka a tieň	2
3.2.	Vykresľovanie minimapy	3
3.3.	Field of View Algoritmus	3
3.4.	Level of Detail Algoritmus	3
3.5.	Animácie	4
3.6.	Loading modelov	4
4.	Ovládanie	4
4.1.	Základné	4
4.2.	Klávesnica	4
4.3.	Konzola	5
5.	Záver	6
6.	Zoznam použitej literatúry	6

1. Predstavenie projektu

Projekt by mal svojim obsahom nadviazať na projekt z premetu PGR, kde sme v teame vytvorili jednoduchú kostru jednoduchej hry. Hra bola pôvodne inšpirovaná hrou Cannon Fodder, z roku 1993, ale počas vývoja sme sa od pôvodných myšlienok odklonili. V rámci predmetu multimédiá som vytvoril multimediálne prvky, išlo predovšetkým o animáciu postavičiek a pridanie zvukových efektov.

Hra je postavená na technológii OpenGL a knižnici GLUT. Využíva aj niektoré prvky multimediálneho systému Windows, ide o MCI. Kód je v jazyku C++ a je preložiteľný pomocou štandardných nástrojov na systémoch Windows (samozrejme s knižnicami GLUT).

2. Zadanie

Zadaním je vytvorenie niekoľkých grafických efektov, zoptimalizovanie vykresľovania a upravenie animácie. Ide o pridanie efektu prechodu slnka a s tým súvisiaceho osvetlenia scény a pridanie tieňov. Ďalej mám za cieľ zoptimalizovať kreslenie minimapy. V rámci optimalizácie ide o pridanie algoritmu na obmedzenie vykresľovania objektov, ktoré sú mimo zorné pole (FOV) a vykresľovania objektov, od kamery vzdialených s nižšou úrovňou detailov (LOD). Ako poslednú úlohu som zvolil rýchlejšie načítanie dátových súborov.

3. Riešenie

3.1. *Efekt slnka a tiene*

Pridanie slnka do projektu nespôsobovalo väčšie problémy. Horšie to bolo s pridaním dynamických tieňov. Pre túto úlohu som zvolil vykresľovanie tieňov pomocou stencil bufferu. Po naštudovaní si problematiky, pre jednoduché objekty vytvorenie tieňov nad scénou nebol väčší problém. Problém nastal až v prípade, keď bolo potrebné aby vrhali tiene modely vytvorené v 3D studiu max

Slnko sa pohybuje okolo scény po kružnici, jeho uhol závisí od uplynutého času. Nastavenie svetiel rozlišuje tri módy. Deň, západ a noc. Počas dňa jedno smerové svetlo osvetľuje scénu a druhé, tmavšie osvetľuje objekty v tieni. Počas západu slnka prechádza svetlo slnka do modrého odtieňu. V noci je svetlo nastavené ako globálne, s modrým nádychom. Tienen sa nevykresľujú.

S vytvorením tieňových objemov (volumes) som si ešte poradil, problém nastal pri ich renderingu, kde sa tieňové volumes prekrývali pre dva proti sebe ležiace trojuholníky. Dobrý príklad je guľa, kde proti sebe ležia dva trojuholníky, pričom pri pohľade z jedného smeru (v tomto prípade zo smeru svetla) je jeden zobrazený prednou stranou a druhý zadnou. Toto spôsobuje problémy, pretože sa tieňové volumes budú u každého vykresľovať v inom poradí (najprv zadné steny a potom predné u prvého trojuholníka a najprv predné a potom zadné u druhého), čo v stencil bufferi spôsobí to, že sa tieto tienie navzájom vrušia (vynulujú). Po dlhom skúmaní problému, som sa dostal konečne na správnu cestu a prišiel na to, že by bolo vhodné pri zobrazovaní tieňových volumes určiť, či je pôvodný trojuholník otočený prednou stranou k svetlu a podľa toho vykresliť tieň. Ideálne je použiť v tomto prípade normálu trojuholníka, ktorá po skalárnom vynásobení s vektorom, ktorý vedie k zdroju svetla,

dostaneme uhol, podľa ktorého sa ľahko určí, či je trojuholník odvrátený, alebo otočený k svetlu.

Väčší problém predstavovalo vytvorenie tieňa animovaných objektov, čo sú práve samotné postavičky. Tento problém sa mi nepodarilo vyriešiť zatiaľ preto sú v projekte použité dynamické tieňe len na stromy.

3.2. Vykresľovanie minimapy

Pri skúmaní prečo vykresľovanie minimapy zaberá toľko výkonu som prišiel na to, že problém bude v rozlíšení textúry minimapy. A teda po upravení rozlíšenia na mocninu dvojky sa minimapa vykresľovala s minimálnym dopadom na výkon. Pre zlepšenie jej výzoru som pridal aj funkciu na spriemerovanie farby pixelu a jeho okolia, takže sa z minimapy odstránili „zuby“.

3.3. Field of View Algoritmus

Algoritmus obmedzenia vykresľovania objektov, ktoré nie sú práve na scéne viditeľné je založený na jednoduchom princípe. Je popísaný nasledovnými krokmi.

- Zistia sa rozmery obrazovky.
- Premietnu sa krajné rohy obrazovky do súradníc mapy (herná plocha so Z súradnicou 0)
- Obalenie vytvoreného štvoruholníka na minimape ohraničujúcim obdĺžnikom (kvôli rýchlosti testu).
- Týmto sa vytvorí obdĺžnik, ktorý zhruba ohraničuje na mape časť, ktorá je zobrazovaná.
- Pri vykresľovaní objektov sa testuje ich poloha, či sa nachádza v tomto obdĺžniku a podľa toho sa vykreslia/nevykreslia.

Algoritmus je jednoduchý a pre úlohy aplikácie postačuje. Nie je ale vylúčené, že sa v budúcnosti zmení na sofistikovanejší a efektívnejší prístup.

3.4. Level of Detail Algoritmus

Keďže Field of View algoritmus nerieši problém kreslenia, keď sa zobrazuje pohľad na celú scénu, nasadzuje sa v tomto prípade Level of Detail algoritmus (LOD). Algoritmus má za úlohy znížiť náročnosť zobrazovania detailných objektov, ktoré sú vzdialené od kamery natoľko, že ich detaily nie sú dobre rozoznateľné. Preto by výsledok tohto algoritmu nemal mať dopad na vizuálnu kvalitu.

Riešenie problému som vyriešil tak, že som v 3D studiu max vytvoril pre každý objekt jeho ekvivalent s nižším počtom polygónov. Celkovo ide o štyri rozdielne náročné na počet trojuholníkov modely pre každý objekt. Tým sa vytvorili štyri úrovne LOD. Jeho aplikácia je založená na vypočítaní vzdialenosti objektu od kamery na základe čoho sa určí zodpovedajúci model, ktorý sa má použiť.

Zmena LOD prebieha automaticky, podľa vzdialenosti kamery od objektu. V projekte je nastavená táto zmena tak, aby bola rozlíšiteľná okom. Pri vypnutí automatického nastavovania LOD je možné meniť detaily objektov ručne pomocou kláves.

3.5. Animácie

Problém animácie spočíval v tom, že pri rýchlom behu hry postavičky prestali vykonávať kroky. Nepriateľské postavičky nevykonávali kroky vôbec.

Po preskúmaní problému som zistil, že nepriateľské postavy mali animáciu v podstate zablokovanú (nastavený veľký časový interval na zmenu kroku animácie). Prvá časť problému spočívala v tom, že animácia bola pevne previazaná s aktuálnym posunom postavy, kde pri veľkých FPS sa postava pohybovala o príliš malé úseky, čo na zmenu animačného kroku nestačilo, preto sa nevykonávala. Zmenu som spravil v tom, že som animáciu dal závislú od akumulovaného posunu. Týmto sa problém úplne vyriešil.

3.6. Loading modelov

V rámci tejto úlohy som sa snažil zrýchliť štart hry, predovšetkým zefektívnením spracovania ASE (exportný ASCII súbor 3D studia max) súborov, z ktorých hra modely a animácie objektov načítava. Preto som zaviedol vlastný formát binárneho súboru, v ktorom sú modely a animácie uložené. Ide prakticky o obsah triedy *cMesh* zapísaný do binárneho súboru.

Hra v prípade ak nemá k dispozícii binárny formát, tak načíta modely a animácie z ASE súborov a hneď ich pre budúce spustenie uloží v binárnom formáte. Rýchlosť nahrávania súborov sa takto zvýšila 80 krát.

4. Ovládanie

4.1. Základné

Počas hry kamera automaticky sleduje pohyb vojaka, preto nie je nevyhnutné jej polohu meniť, je to však možné, stlačením jedného z vhodných klávesov a pohybom myši.

Vojaci sa ovládajú klikaním ľavým tlačítkom myši (LMB) do mapy, a na danú polohu sa vojaci snažia presunúť. Pri stlačení pravého tlačítka (RMB), vojaci či už sú v pohybe, alebo nie začnú strieľať v smere kurzora.

Granáty, alebo rakety sa hádžu, resp. strieľajú stlačením ľavého tlačítka, pri stlačení pravom. Prakticky to znamená, že vojak hodí granát vtedy, ak užívateľ stlačí ľavé tlačítko počas streľby. Granát hádžu vojaci na polohu kurzora (ak dohodia), a rakety strieľajú v smere kurzora (aj raketa má obmedzená dolet).

4.2. Klávesnica

Ovládanie pomocou klávesnice je v nasledovnej tabuľke. Nezahŕňa všetky možné príkazy konzole (odetrovanie prázdneho riadku vypíše príkazy), ale zahŕňa testovacie klávesy, ktoré slúžia na testovanie hry (developer mode), ale s hrou nesúvisia vôbec, sú označené červenou farbou. Ovládanie klávesnicou ako také, nie je v hre potrebné (až na klávesy ENTER a ESC, k stlačeniu ktorých ale hra priamo vyzýva).

klávesa/vstup	funkcia
[F1]/[F2]	zmeny priehľadnosti konzoly
[F3]	prepínanie priehľadnosti fps
[F5]	zmena na režim plného výkonu (okno sa prekresľuje ako najčastejšie je to možné)
[~]	zobrazenie/ukrytie konzoly
[t]	zmena z granátov na rakety a naopak
[p]	pauznutie hry
[ESC]	ukončí level a vráti sa do menu
[1] + pohyb myši	rotácia kamery okolo osi X
[2] + pohyb myši	rotácia kamery okolo osi Z
[3] + pohyb myši	zoom kamery
[medzerník] + pohyb myši	posúvanie kamery nad terénom
[stredné tlačítko myši]	skok do ďalšieho levelu
[v]/[b]	zvýšenie/zníženie LOD úrovne (menej je lepší model, použiteľné len pri vypnutom LOD)
[g]	zapnutie/vypnutie automatického LOD
[d]	zapnutie/vypnutie denného cyklu
[f]	zapnutie/vypnutie dynamických tieňov
[+]/[-]	zvyšovanie/znižovanie počtu tieniacich stromov (zvyšuje výkon)
[l]	zapnutie/vypnutie lifemeters
[m]	zapnutie/vypnutie FOV
[n]	zapnutie/vypnutie minimapy
[a]/[s]	pridanie/odobratie nepriateľa
[q]/[w]	pridanie/odobratie hráčovho vojaka
[k]	zraní všetkých nepriateľov

4.3. Konzola

Konzola predstavuje jednoduchý systém na nastavovanie parametrov hry počas behu, bez potreby pamätať si nespočetné množstvo kláves.

Konzola sa vyvolá klávesou „~“ (česky tilda, slovensky vlnka). Keď sa „odentruje“ prázdny riadok konzola vypíše všetky príkazy, ktoré podporuje. Za zmienku stojí príkaz „controls“, ktorý vypíše ovládanie hry. Špeciálny je príkaz „god“ s parametrom 0, alebo 1, ktorý zapína mód nesmrteľnosti, inými slovami tzv. cheat.

Konzola je stále pod vývojom, a pribúdajú v nej nové pekné vlastnosti ako automatické dopĺňanie príkazov, alebo možnosť listingu.

5. Záver

Pri projekte som pracoval s projektom po dlhšom čase, dobrý pocit som mal z toho, že som sa v kóde rýchlo zorientoval, iný človek by s tým mal ale asi väčší problém. Implementácia niektorých vlastností si vyžiadala podstatné zmeny v kóde, niektoré z nich sa nepodarilo dokončiť, ako napr. dynamické tieňe pre animované objekty. Toto by ale vyžadovalo podstatnú zmenu animačného systému (v používanom modeli je to možné implementovať, ale s kritickým dopadom na výkon).

Jozef Radakovič
Brno, 15.január 2008

6. Zoznam použitej literatúry

[1] Wikipedia *Cannon Fodder* [online]. Posledná modifikácia 2008. Dostupné na URL: http://en.wikipedia.org/wiki/Cannon_Fodder

[2] Wikipedia *Shadow volume* [online]. Posledná modifikácia 2008. Dostupné na URL: http://en.wikipedia.org/wiki/Shadow_volume

[3] online skriptá, Katedra matematiky a deskriptívnej geometrie Stavebnej Fakulty STU, Bratislava, <http://www.math.sk/skripta/node22.html>

[4] online skriptá, <http://www.sweb.cz/radek.jandora/f12.htm>

[5] Žára, J., Beneš, B., Sochor, J., Felkel, P. *Moderní počítačová grafika*. Computer Press, Brno, 2004. ISBN 80-251-0454-0