



ZÁPADOČESKÁ
UNIVERZITA
V PLZNI

Semestrální práce

z předmětu

Základy počítačové grafiky

Jméno a příjmení: Martin Sloup
Osobní číslo: A04372
Datum odevzdání: 3. listopadu 2006
E-mail: mssloup@students.zcu.cz

Zadání

Vytvořte herní engine ve stylu Wolfenstein 3D.

Implementace

Herní engine jsem vytvořil v programovacím jazyce C# za pomoci vývojového prostředí Visual Studio 2005 a Managed DirectX SDK ve verzi ze srpna 2005.

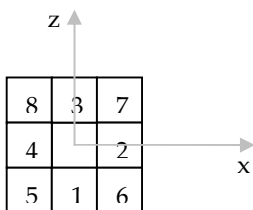
Ovládání

Herní engine Blue-dish-te se ovládá podle zadání, které jsem dostal, tj.:

- Klávesy W, S, A, D pro pohyb vpřed, vzad, úkok vlevo a úkok vpravo
- Klávesa U zapíná invertaci vertikální osy u myši
- Dále vlastní klávesa L umožňuje načíst jinou mapu, tím že otevře dialog pro vybrání souboru s mapou (toto je dočasné řešení, které bude odstraněno použitím konzole známé například ze hry Quake III)
- Vlastní klávesa H nastavuje hráče zpět na startovní pozici
- Vlastní klávesa K umí uvolnit myš a povolit například změnu velikosti okna, či provádět jinou práci při spuštění engine
- Pohybem myši nahoru / dolů, doleva / doprava, s možností invertace vertikální osy

Zpracování kolizí

O zpracování kolizí se stará třída `kolize`, ve které se, po zpracování zmáčknutých kláves, zavolá funkce `upravPohyb`. Této funkci se předá poloha hráče před a po zpracování zmáčknutých kláves. Nejprve se zkontroluje, zda se hráč snaží vypadnout z mapy, pokud se tak děje, nastavíme ho 0.5f od okraje (0.5f je tam definováno pomocí konstanty `hracRadius`)



Nyní vezmeme starou pozici hráče a otestujeme ve vodorovném a svislém směru zda následující či předchozí kostička obsahuje stěnu (dle přiloženého obrázku stará pozice hráče je počátek os a testujeme na mapě kostičky 1, 2, 3 a 4). Pokud tomu tak je a zároveň nová poloha hráče se již nachází v příslušné testované kostičce či za ní (s respektováním příslušné mezery `hracRadius` od začátku stěny), nastavíme novou polohu hráče vrácením hráče směrem ke staré poloze o jednu kostičku od testované kostičky společně s posunutím tak, aby se hráč nacházel o `hracRadius` od stěny. Testováním kostiček 5, 6, 7 a 8 předejdeme problémům na rozích v mapě.

Obecně tedy splněním podmínky (nová poloha hráče se již nachází v příslušné testované kostičce či za ní) při testování kostiček:

- 5, 1 a 6 nastavujeme $nové.z = testovana_kosticka.z + 1 (- hracRadius)$
- 6, 2 a 7 nastavujeme $nové.x = testovana_kosticka.x - 1 (+ hracRadius)$
- 8, 3 a 7 nastavujeme $nové.z = testovana_kosticka.z - 1 (+ hracRadius)$
- 8, 4 a 5 nastavujeme $nové.x = testovana_kosticka.x + 1 (- hracRadius)$

Pozn.: Víšše uvedené `hracRadius` v závorce jen ukazuje, jak má být provedeno odsazení od stěny (příslušné odečtení). Ve výsledném kódu je to trochu složitější, jelikož je nutné převádět mezi mapou, kde jedna kostička zabírá 1 (typu int) a interpretací ve 3D prostoru, kde jedna kostička má šířku či délku 2 floaty.

Nakonec funkce vrací novou polohu hráče na mapě.

Součásti semestrální práce

- Zdrojové kódy a projekt VS 2005
- tato dokumentace ([dokumentace-prvni-odevzdani.pdf](#))
- adresář s texturami ([adresář textury](#))
- různé mapy s příponou `map` (engine vyžaduje pouze mapu `semestra1ka.map`)

Pozn.: Pro správný běh engine je nutné mít v adresáři, kde se nachází spustitelný program i adresář `textury` a mapu `semestra1ka.map`, jinak nelze engine spustit. Adresář s texturami se nachází i v adresáři `bin/Debug`, aby bylo možné spustit engine přímo z vývojového prostředí VS 2005.