

ZÁPADOČESKÁ  
UNIVERZITA  
V PLZNI

# Semestrální práce

z předmětu

## Základy počítačové grafiky

Jméno a příjmení: Martin Sloup  
Osobní číslo: A04372  
Datum odevzdání: 18. prosince 2006  
E-mail: [msloup@students.zcu.cz](mailto:msloup@students.zcu.cz)

## Zadání

Vytvořte herní engine ve stylu Wolfenstein 3D.

## Programátorská dokumentace

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

## Struktura programu

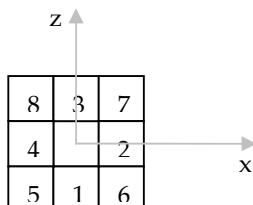
- **RenderForm**
- **Svet**
- **Kolize**
- **Strop**
- **Podlaha**
- **Stena**
- **MapObject**
- **Teleport**
- **Dvere**
- **LahevObject**
- **ChipsyObject**

## Popis tříd

- **RenderForm**  
Třída zastřešující celý engine
- **Svet**  
Třída starající se kreslení grafických objektů a práci s mapovými podklady
- **Kolize**  
Třída řešící detekci kolizí
- **Strop**  
Třída renderující strop
- **Podlaha**  
Třída renderující podlahu
- **Stena**  
Třída renderující stěnu v bludišti
- **MapObject**  
Výchozí třída která zastřešuje objekty, od ní dědí následující objekty
- **Teleport**  
Třída starající se o teleport.
- **Dvere**  
Třída starající se o dveře.
- **LahevObject**  
Třída kreslící objekt pivní láhve.
- **ChipsyObject**  
Třída kreslící objekt chipsů.

## 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 hracRadi us 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 nove. z = testovana\_kosti cka. z + 1 ( - hracRadi us)
- 6, 2 a 7 nastavujeme nove. x = testovana\_kosti cka. x - 1 ( + hracRadi us)
- 8, 3 a 7 nastavujeme nove. z = testovana\_kosti cka. z - 1 ( + hracRadi us)
- 8, 4 a 5 nastavujeme nove. x = testovana\_kosti cka. x + 1 ( - hracRadi us)

Pozn.: Víšše uvedené hracRadi us 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.

## ***Kreslení jen té zdi, která je vidět***

Třída Svet při načítání mapy prochází celou mapu a zjišťuje, které zdi jsou vidět. Takže nakonec se vykresluje jen to, co může hráč procházející mapou vidět. Zdi vykresluje pouze jeden objekt. To proto, aby se nealokovalo tolik objektů v paměti.

## ***Uživatelská dokumentace***

Ovládání je velice jednoduché a vyhovuje všem standardním zvyklostem 3D her.

## ***Ovládání***

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

- Klávesy „W“, „S“, „A“, „D“ pro pohyb vpřed, vzad, úkrok vlevo a úkrok vpravo
- Klávesa „U“ zapíná invertaci vertikální osy u myši
- Klávesa „E“ nebo „Ester“ otevírají dveře
- 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“ či „R“ 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
- Klávesa „Tab“ zobrazuje / schovává nápovědu
- Klávesy pro změnu režimu vykreslování:
  - Klávesa „1“ pro zapnutí/vypnutí speciálního drátěného modelu
  - Klávesa „2“ pro zapnutí/vypnutí světél
  - Klávesa „3“ pro zapnutí/vypnutí mlhy
  - Klávesa „4“ pro zapnutí/vypnutí vykreslování textur
- Pohybem myši nahoru / dolů, doleva / doprava, s možností invertace vertikální osy

## ***Variace***

### **Drátěný model (2 b)**

Zobrazení drátěného modelu scény s respektováním viditelnosti jednotlivých hran. Čárový mód nesmí ovlivnit zobrazení UI (User Interface) a HUD (Heads Up Display).

Drátěný model se vykresluje dle zadání a je přístupný přes klávesu „1“. Při drátěném modelu se musí scéna vykreslit dvakrát, jednou jen do z-bufferu v solid módu, čímž se nastaví, co bude a nebude vidět a po druhé jako drátěný model.

### **Textura (5 b)**

Načtení a aplikace textury s vizuálně nejlepším možným způsobem vzorkování textury dostupným na daném HW. Úloha předpokládá vytvoření autodetekce pro detekování nejlepšího přístupu jaký je daným HW podporován: nearest-neighbor, linear, anisotropic, nearest-neighbor mip-map, linear mip-map (též trilineární filtrování), atd.

Program při startu detekuje schopnosti grafiky a nastaví příslušné nastavení.

## **Sběr předmětů (5 b)**

V mapě rozložený předměty. Hráč předměty sbírá a počítá se počet sebraných předmětů, který je zobrazen na obrazovce. Předměty poskakují, točí se na místě, popř. provádějí nějaký jiný zajímavý pohyb. Rychlost pohybu předmětů je nezávislá na snímkovém kmitočtu.

Do mapy jsem umístil dva předměty pro sběr: flašku piva a sáček chipsů. Pokud hráč vypije moc piva, začíná se motat, a po vypití přílišné dávky umírá. Toto je zobrazeno opilostí v procentech. Opilost lze snížit sebráním sáčku chipsů. Objekty jsem řešil vymodelováním jako mesh objekty v programu Anim8or.

## **Teleport (5 b)**

Teleportace z místa na místo. Implementace efektu pro hráče: přechod obrazu do bíla/z bílé do cílové scény.

Teleport je v mapě označen znakem „+“. Teleportace se spustí, pokud hráč vstoupí na teleport. Následně se provede přechod do bíla, nastavení nové pozice hráče na následující teleport v mapě a přechod zpět do cílové scény. Tento přechod je řešen pomocí mlhy implementované v Direct X.

## **Dveře a tajné chodby (5 b)**

Animace otvírání + interakce s hráčem (otevřeno/zavřeno/mění stav). Postačuje animace na úrovni Wolfenstein 3D.

Dveře zajiždějí do zdi a po čase se automaticky sami zavírají, pokud nestojí hráč uprostřed dveří, v tom případě se vynuluje časovač a čeká se dalších 10 vteřin, než se zavřou dveře.

## **Součástí semestrální práce**

- Zdrojové kódy a projekt VS 2005
- tato dokumentace (dokumentace-posl edni -odevzdani . pdf)
- soubor s texturami a objekty (soubor blue-dish-te.dat)
- různé mapy s příponou map (engine vyžaduje pouze mapu semestral ka. map)

Pozn.: Pro správný běh engine je nutné mít v adresáři, kde se nachází spustitelný program i soubor blue-dish-te.dat a mapu semestral ka. map, jinak nelze engine spustit. Soubor blue-dish-te.dat se nachází i v adresáři bin/Debug, aby bylo možné spustit engine přímo z vývojového prostředí VS 2005.

## **Závěr**

Během vývoje tohoto engine jsem se naučil základů práce s grafikou v Managed Direct X. Pro mě byla práce přínosná, protože jsem si chtěl alespoň jednou napsat 3D hru, či aspoň její část. Při práci na engine jsem se seznámil i s programovacím jazykem C# a vývojovým prostředím Microsoft Visual Studio 2005, ve kterém bych chtěl nadále pracovat, jelikož považují jak C# tak i samotný IDE jako jednu z nejlepších kombinací pro vývoj SW.

## **Použité zdroje a programy:**

- Microsoft Visual Studio 2005 + Direct X SDK August 2005
- Odkazy z [http://en.wikipedia.org/wiki/Direct\\_x#Programmer\\_resources](http://en.wikipedia.org/wiki/Direct_x#Programmer_resources)
- Návod y z <http://herakles.zcu.cz/education/zpg/navody.php>
- Program Anim8or na modelovani pohybujících se objektů (<http://www.anim8or.com/>)