Základy Počítačové grafiky

Tomáš Maršálek

27. října 2012

1 Úvod

Pro semestrální práci jsem po osobní dohodě se cvičícím zvolil jazyk Java, poté podle vlastního výběru jeden z nejpoužívanějších wrapperů OpenGL pro Javu - **Lightweight Java Game Library (LWJGL)**.

2 Výpočty

2.1 Terén

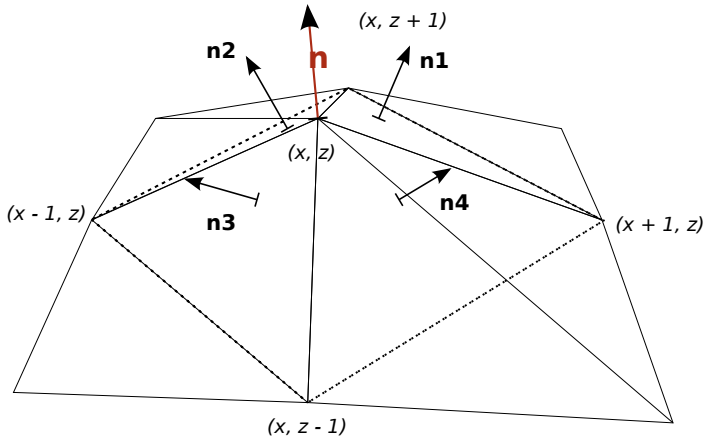
Terén je načten ze souboru a v programu používán jako objekt třídy **Terrain**. Terén je $H-1$ trojúhelníkových pruhů (*GL_TRIANGLE_STRIPS*) délky W. Objekt **Terrain** ulehčuje práci pomocí metod umožňujících získání výšky nebo normály v daném bodě této trojúhelníkové sítě.

2.2 Výpočet normál terénu

Normála v každém bodě trojúhelníkové sítě je vypočtena jako průměr normál čtyř okolních myšlených trojúhelníků, popřípadě tří normál na hraně terénu nebo dvou v rozích.

$$\begin{aligned}
\mathbf{n1} &= [\mathbf{terrain}(x+1,z) - \mathbf{terrain}(x,z)] \times [\mathbf{terrain}(x,z+1) - \mathbf{terrain}(x,z)] \\
\mathbf{n2} &= [\mathbf{terrain}(x,z+1) - \mathbf{terrain}(x,z)] \times [\mathbf{terrain}(x-1,z) - \mathbf{terrain}(x,z)] \\
\mathbf{n3} &= [\mathbf{terrain}(x-1,z) - \mathbf{terrain}(x,z)] \times [\mathbf{terrain}(x,z-1) - \mathbf{terrain}(x,z)] \\
\mathbf{n4} &= [\mathbf{terrain}(x,z-1) - \mathbf{terrain}(x,z)] \times [\mathbf{terrain}(x+1,z) - \mathbf{terrain}(x,z)]
\end{aligned}$$

$$\begin{aligned}
\mathbf{n} &:= \mathbf{n1} + \mathbf{n2} + \mathbf{n3} + \mathbf{n4} \\
\mathbf{n} &:= \frac{\mathbf{n}}{\|\mathbf{n}\|}
\end{aligned}$$



2.3 Detekce kolizí při pohybu po terénu

Nejprve je spočten směrový vektor v rovině XZ podle momentálně stisknutých kláves WSAD. Výšková složka tohoto vektoru je dopočtena jako rozdíl výšky v terénu nad současnou pozicí a výškou nad novou pozicí určenou směrovým vektorem v rovině. Vektor je pak normalizován a zvětšen rychlostním koeficientem, který zaručuje, že rychlost bude konstantně 3m/s nehledě na snímkové frekvenci.

Tento rychlostní vektor je pak přičten k současné pozici.

Problém však nastává s výškou pozorovatele nad terénem, která je pevně dána na 1.85m. Použitím relativního vektoru rychlosti dochází ke kumulaci zaokrouhlovacích chyb a po chvíli chození po terénu je výška očí zcela jiná, než 1.85m. Proto je v každém snímku provedena korekce, která zaručuje, že oči budou neustále stejně vysoko.

2.4 Plynulá intenzita světla

Máme-li den, který trvá od času 0 (00:00) do 1 (24:00), očekáváme v čase 0.5 nejvyšší intenzitu světla. Použitá funkce určující intenzitu slunce je Gaussova křivka centrovaná do bodu 0.5.

$$Intenzita \ = \ e^{-(C(t-\frac{1}{2}))^2}$$

kde C je magická konstanta, která byla nalezena po chvíli testování tak, aby byl výsledek co nejbližší skutečnosti.

Získaná intenzita se používá přímo jako intenzita hlavního zdroje světla - slunce a jako barva atmosféry.

3 Přeložení a spuštění

Aplikace byla vyvíjena pod GNU/Linux, přiložené knihovny a instrukce pro spuštění aplikace jsou pro Windows.

3.1 Adresářová struktura

3.2 Z příkazové řádky

přeložení:

```
$  javac -cp .;lib\jars\lwjgl.jar:lib\jars\lwjgl_util.jar; src\Main.java
src\Terrain.java
```

spuštění:

```
$  java -cp .;lib\jars\lwjgl.jar;lib\jars\lwjgl_util.jar; -Djava.library.path=lib/natives-win
```

3.3 Použitím Ant

Pokud je nainstalovaný Apache Ant, pak pro spuštění stačí příkaz

```
$  ant
```

3.4 V eclipse

Získání a integrace **LWJGL** pod eclipse je popsána ve videu [2], ale není třeba. Práce je odevzdána přímo jako projekt eclipse, tedy stačí vytvořit projekt Z

Reference

[1] *Bill Jacobs*
OpenGL tutorial 2007 - 2012
`http://www.videotutorialsrock.com/`

[2] *TheCodingUniverse*
#1 LWJGL Workspace - LWJGL Tutorials
2011
`http://youtu.be/0v56I5UWrYY`