



**Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky
Centrum počítačové grafiky a vizualizace dat
Základy počítačové grafiky KIV/ZPG**

Průchod terénem (Passage of terrain)

Zadání:

Zadáním semestrální práce je vytvoření aplikace pro průchod terénem na smyšlené "planetě". Aplikace bude kompletně provedena v prostředí .NET s využitím rozhraní OpenGL.

Terén bude zadán polem alespoň 128x128 výšek a vykreslený pomocí trojúhelníkové sítě. Výšková mapa terénu bude načtena z binárního souboru, jedna výška je zde reprezentována jedním bytem, výšky jsou uloženy v řádcích od západu k východu, řádky jsou uloženy od severu k jihu.

Průchod pozorovatele terénem je ve výšce očí (1,85 m), výšku terénu mezi body zadanými výškovou mapou je třeba interpolovat. Pozorovatel nesmí vylézt z konce výškové mapy a za žádných okolností nesmí vidět nebo vstoupit do terénu. Pozorovatel se pohybuje konstantní rychlostí (3 m/s). Pozorovatel se může pohybovat pomocí klávesnice a rozhlížet se pomocí myši (nesmí se však přetočit při pohybech nahoru/dolů, rozsah pouze (-90° ; +90°)). Výchozí pozice pozorovatele je uprostřed mapy.

Naše "planeta" je osvětlena sluncem, jehož doba oběhu je 2 minuty (tj. "světlo" je jednu minutu, "tma" také jednu minutu). Přechod mezi dnem a nocí musí být plynulý. Během dne se mění intenzita slunečního svitu a uzpůsobuje se i barva atmosféry.

Prohlášení:

Prohlašuji, že všechny použité zdroje jsou řádně citovány a předkládaná práce byla vytvořena zcela mnou, pokud není uvedeno jinak.

e-mail studenta:	jmoulis@students.zcu.cz	Datum:	25. 10. 2011
Kód studenta:	A09B0359P		
Jméno studenta:	Jan Moulis	Hodnocení (počet bodů):	
Datum odevzdání:	27. 10. 2011	Jméno cvičícího:	Ing. Oldřich Petřík
Podpis studenta:		Podpis cvičícího:	

1. Úvod

1.1. Problematika

Při průchodu výškovou mapou se musí řešit kolize pozorovatele s terénem (pozorovatel musí chodit ve výšce 1,85 m nad terénem a nesmí vstoupit do terénu).

Dalším problémem je rychlost pohybu pozorovatele (3 m/s) a doba oběhu slunce (2 min = 3° /s). Oba tyto pohyby musí být závislé na reálném čase a ne na snímkovém kmitočtu.

Jako další problematiku bych uvedl intenzitu slunečního svitu a změnu barvy atmosféry. Oba tyto problémy jsou závislé na aktuální pozici slunce.

1.2. Řešení

Řešením kolize pozorovatele s terénem je Bilineární interpolace, pomocí které dokážeme určit přibližnou hodnotu výšky terénu v daném bodě.

Rychlost pohybu pozorovatele a dobu oběhu slunce, musíme počítat v závislosti na době, která uběhne mezi jednotlivým rendrováním a dopočítávat z této doby posun pozorovatele a pozici slunce.

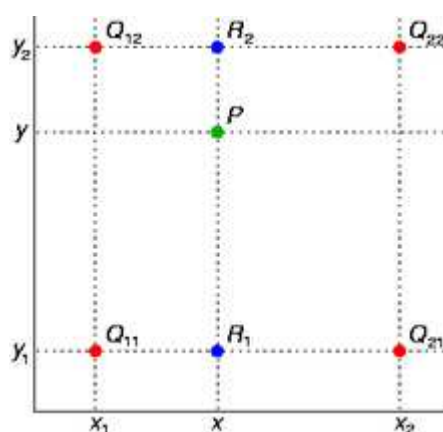
Sluneční svit i změna barvy atmosféry je řešena vesměs stejným způsobem, založeným na úhlu, ve kterém se nachází slunce.

2. Známé metody

2.1. Výpočet výšky a Bilineární interpolace

Pomocí Bilineární interpolace jsme schopni vypočítat přibližnou výšku terénu v kterémkoliv místě a následně upravit pozici pozorovatele do správné výšky nad terén.

Bilineární interpolace je rozšíření lineární interpolace pro interpolaci funkce dvou proměnných na pravidelnou prostorovou mřížku. Klíčová myšlenka je provést lineární interpolaci nejprve v jednom směru a pak i ve druhém směru. Tím dostaneme odhad $f(x, y)$.



Obr. 1: Body pro Bilineární interpolaci

Červené body Q11, Q21, Q12 a Q22 na Obr. 1: Body pro Bilineární interpolaci jsou výškové body načtené z binárního souboru. X a Y je pozice bodu P, ve kterém chceme spočítat přibližnou hodnotu funkce, v našem případě výšku v tomto bodě.

$$\begin{aligned}
f(x, y) \approx & \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y_2 - y) \\
& + \frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y_2 - y) \\
& + \frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y - y_1) \\
& + \frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y - y_1).
\end{aligned}$$

Obr. 2: Vzorec Bilineární interpolace

Podle vzorce na Obr. 2: Vzorec Bilineární interpolace vypočteme přibližnou hodnotu v bodě P a následně k této výšce přičteme výšku postavy pozorovatele 1,85 metru a do této výsledné výšky přesuneme pozorovatele.

Při reálném výpočtu v programu je nejprve nutné zjistit pozici pozorovatele, a jelikož je výškový terén vzorkován po 2 metrech, je nutné přepočítávat tyto souřadnice, abychom byli schopni získat z pole výšek výšky čtyř vrcholů čtverce. Vzorec Bilineární interpolace se zjednoduší, jelikož víme, že rozměr jednoho čtverce je vždy 2x2 metry.

Výpočet výšky v programu:

```

int x2 = getPositionInField(posX);           // zjištění x pozice bodu Q11
int z2 = getPositionInField(posZ);           // zjištění z pozice bodu Q11
float xz = pole[z2][x2];                     // výška v bodu Q11
float x1z = pole[z2][x2 + 1];                // výška v bodu Q21
float xz1 = pole[z2 + 1][x2];                 // výška v bodu Q12
float x1z1 = pole[z2 + 1][x2 + 1];            // výška v bodu Q22
float xPos = - posX - 2 * x2;                 // zjištění x souřadnice bodu P
float zPos = - posZ - 2 * z2;                 // zjištění z souřadnice bodu P
vyska = xz / 4 * (2 - xPos) * (2 - zPos)      // vzorec Bilineární interpolace
        + x1z / 4 * (xPos) * (2 - zPos)      // výpočet přibližné výšky y
        + xz1 / 4 * (2 - xPos) * (zPos)
        + x1z1 / 4 * (xPos) * (zPos);
posY = -vyska - velikostPostavy;              // nastavení výšky pozorovatele

```

2.2. Rychlost pozorovatele a doba oběhu slunce

Jak již bylo napsáno v úvodu, rychlost pohybu pozorovatele a dobu oběhu slunce, musí být stále stejná a nesmí záviset na snímkovacím kmitočtu. Proto je nutné zjišťovat si přesný čas před každým rendrováním a následně spočítat dobu od posledního rendrování. Když známe dobu mezi rendrováním, můžeme následně posouvat pozorovatele o přesně danou dráhu, kterou ušel za tuto dobu rychlostí 3 m/s.

Stejný postup je u slunce, jen s tím rozdílem, že rychlost slunce je dána dobou oběhu planety. Tato doba je 2 minuty. Z této doby jsme schopni dopočítat, že za 1 sekundu se slunce pohne o 3°. Jeho obvodová rychlost je tedy 3°/s.

2.3. Sluneční svit a změna barvy atmosféry

Pro výpočet slunečního svitu a změny barvy atmosféry je zde použita funkce cosinus, která nám z úhlu, ve kterém se nachází slunce, spočítá hodnotu svitu a hodnotu modré složky atmosféry.

Výpočet slunečního svitu a modré složky atmosféry v programu:

```
blue = Math.Abs((float)Math.Cos(sunAngle * rad)); // výpočet barvy atmosféry
light = blue + 0.5f; // výpočet intenzity světla
gl.Lightfv(GL.LIGHT1, GL.DIFFUSE, new float[] { light, light, light, 1.0f });
// nastavení osvětlení
gl.ClearColor(0, 0, blue, 0); // změna barvy atmosféry
```

3. Závěr

Základní funkcionalita programu je plně funkční podle rozsahu zadání.

Nad rámec rozsahu je v programu doděláno otexturování povrchu texturou egyptské pouště, dále je přidána krychle jako "slunce", aby měl pozorovatel možnost zjistit jaká je doba dne a mohl se zorientovat. Toto "slunce" je také otexturováno. Dále je v programu možnost navrátit se do výchozí pozice (středu mapy) a přepínat mezi zobrazením ploch a zobrazením hran a (viz. Ovládání programu).

Některé problémy jsem řešil a konzultoval s ostatními studenty:

A09B0411P Václav Rajtmajer

A09B0415P Kamil Rendl

A09B0417P Jakub Rinkes

4. Literatura

Zadání semestrální práce

<http://herakles.zcu.cz/education/zpg/cviceni.php>

NeHe Neon Helium Production vše o programování 3D grafiky s knihovnou OpenGL

NeHe OpenGL tutoriály

http://nehe.ceske-hry.cz/tut_obsah.php

Projekty a zdrojové kódy Lodherov21

Hawaii OpenGL (v01)

<http://www.lodherov21.eu/projekty/pages/programovani-zdrojove-kody-opengl.html>

Diskusní forum Builder

<http://forum.builder.cz/>

OpenGL Super Bible

http://herakles.zcu.cz/local/manuals/OpenGL_SuperBible/ewtoc.html

Wikipedia Bilineární interpolace

http://cs.wikipedia.org/wiki/Biline%C3%A1rn%C3%AD_interpolace

5. Ovládání programu

myš	otáčení vlevo/vpravo, rozhlížení nahoru/dolů (pohyb myši vpřed = nahoru)
klávesa "W"	pohyb kupředu ve směru pohledu
klávesa "S"	pohyb vzad vzhledem ke směru pohledu
klávesa "A"	pohyb vlevo vzhledem ke směru pohledu (nikoliv otáčení)
klávesa "D"	pohyb vpravo vzhledem ke směru pohledu (nikoliv otáčení)
klávesa "U"	zapnutí/vypnutí invertace myši pro pohled nahoru/dolů.
klávesa "F"	přepnutí mezi zobrazením ploch a hran
klávesa "R"	navrácení se do výchozí pozice