

**Západočeská univerzita v Plzni**  
**Fakulta aplikovaných věd**  
**Katedra informatiky a výpočetní techniky**  
**Centrum počítačové grafiky a vizualizace dat**

Předmět: KIV/ZPG

## ZPG vstupní test

**Zadání úlohy:** viz příloha č. 1: papers/priloha-1-zadani.pdf

**Prohlášení:**

Prohlašuji, že všechny použité zdroje jsou řádně citovány

e-mail studenta: neumann@students.zcu.cz	
Jméno a příjmení studenta: Antonín Neumann	Hodnocení (počet bodů):
Datum odevzdání: 4. 11. 2013	Jméno a příjmení vyučujícího: Ing. Michal Šmolík
Podpis studenta:	Podpis vyučujícího:

## 1. Úvod

V první části samostatné práce bylo úkolem načíst ze vstupního souboru profil tělesa a k němu vygenerovat množinu bodů definujících jeho povrch. Dále bylo požadováno vytvoření trojúhelníkové sítě tělesa s využitím datové struktury okřídlená hrana a její vykreslení pomocí implicitních metod.

V druhé části poté bylo požadováno zapsání okřídlené hrany v požadovaném formátu do výstupního souboru. A vykreslení trojúhelníkové sítě pomocí vlastních matic.

## 2. Generování bodů

K vygenerování množiny bodů tvořících povrch tělesa, jsem využil metodu, kterou jsem implementoval již ve své první samostatné práci. Tato metoda vypočte rovnoměrné rozložení bodů na kružnici.

Při implementaci této metody ve 3D prostředí, si bylo nutné uvědomit, že rotace probíhá kolem osy Y.

## 3. Trojúhelníková síť

Při implementaci výpočtu trojúhelníkové sítě bylo potřeba se zamyslet na způsobem výpočtu. Takže jsem si povrch představil jako plochu ve 2D. Nejprve jsem vytvořil čtverce spojením dvou sousedních bodů z jedné úrovně (úroveň je myšlena kružnice bohů vytvoření z každého původní bodu profilu tělesa) se dvěma odpovídajícími sousedními body z nižší úrovně.

Následně jsem každý čtverec rozdělil úhlopříčkou na dva trojúhelníky. Jeden trojúhelník jsem nazval sudý (ten jehož index v poli trojúhelníků byl sudé číslo) a druhý lichý. Pro každý trojúhelník probíhal způsob výpočtu nepatrně odlišným způsobem.

Největší pozor je třeba si dát na poslední trojúhelník v řádce, který se opět napojuje na první trojúhelník dané řádky a dochází tedy k návratu v indexaci pole.

Po vytvoření všech trojúhelníků bylo zapotřebí k jejich vykreslení dopočítat normálový vektor příslušející každému trojúhelníku. Tento se vypočítat velmi snadno jako vektorový součin vektorů tvořících dvě přilehlé strany trojúhelníka. Tyto vektory se dají snadno vypočítat jako rozdíl vzdáleností dvou bodů.

Nakonec, aby bylo možné celou trojúhelníkovou síť zapsat do jako datovou strukturu okřídlená hrana, bylo nutné vypočítat pro každý trojúhelník jeho sousední trojúhelníky. Při tomto výpočtu je třeba si dávat pozor hlavně u první a poslední vrstvy, protože po ní resp. před ní žádná další vrstva neexistuje a tedy zde není ani žádný sousední trojúhelník. Nastane-li tento stav, zapíše se místo indexu sousedního trojúhelníka číslo -1.

Takto vytvořenou datovou strukturu zapíšeme do souboru podle předem daného schématu, jenž je definován v zadání. Viz příloha číslo 1.

## 4. Vlastní vykreslení trojúhelníkové sítě

Posledním úkol bylo vykreslit těleso osvětlené jedním zdrojem světla, v náklonu  $30^\circ$  od osy Y, dále konstantní úhlovou rychlostí  $\omega_2$  kolem osy Y a konstantní úhlovou rychlostí  $\omega_1$  kolem své vlastní středové osy.

Požadavkem také bylo aby rychlost otáčení nebyla závislá na výkonu počítače, ale byla konstantní. K tomu je zapotřebí použití systémových hodin.

Všechny rotace je nutné počítat pro těleso v základní poloze, tudíž každý 10ms vrátíme těleso do původního stavu a opětovně provedeme výpočet matic rotace a tyto matice poté sečteme do výsledné transformační matice, kterou následně aplikujeme na naše těleso.

## 5. Závěr

Práce mi ukázala základní pohled na vykreslování objektů ve 3D prostoru pomocí trojúhelníkové sítě, která je v současné době asi nepoužívanější metodou pro vykreslování renderování 3D scén.

Také jsem si vyzkoušel aplikovat na vytvořené těleso několik transformací najednou.

Vzhledem k předem požadované datové struktuře okřídlená hrana jsou i paměťové nároky aplikace nižší než by mohli být v případě volné volby datové struktury. Přesto si myslím, že by některé výpočty

šlo optimalizovat. Například výpočet normálového vektoru trojúhelníka by šlo počítat souběžně s jeho vytvářením a ne odděleně v samostatné metodě jak to mám v současné verzi implementováno.

## 6. Literatura

### Publikace WEB

[Tran01] Transformace I. *Ing. Michal Šmolík* [online]. [cit. 2013-11-4]. Dostupné z:

[http://home.zcu.cz/~smolik/zpg/cviceni/cv\\_03.html](http://home.zcu.cz/~smolik/zpg/cviceni/cv_03.html)

[Tran02] Transformace II. *Ing. Michal Šmolík* [online]. [cit. 2013-11-4]. Dostupné z:

[http://home.zcu.cz/~smolik/zpg/cviceni/cv\\_04.html](http://home.zcu.cz/~smolik/zpg/cviceni/cv_04.html)