

GRAFICKÉ INTRO 64KB S POUŽITÍM OPENGL

Zdeněk Hejl

Bachelor Degree Programme (3), FIT BUT

E-mail: xhejlz00@stud.fit.vutbr.cz

Supervised by: Herout Adam

E-mail: herout@fit.vutbr.cz

ABSTRACT

This document systematically describes main parts of implementation and used techniques in a graphic intro and explains how some effects were achieved. There is described how object like terrain, trees, bush, grass and others are generated, textured and optimized.

1. ÚVOD

Cílem této bakalářské práce je vytvořit krátké ozvučené video s co nejhezčími scénami za použití minimalistických technik, umožňujících výslednou velikost do 64kB.

2. ROZBOR

2.1. OPENGL APLIKACE

Prvním krokem ve vývoji bylo vytvořit OpenGL okno za pomoci co nejmenšího počtu knihoven, které by zbytečně zvětšovaly velikost výsledné aplikace. Pro testovací účely bylo třeba vytvořit základní prostředky pro orientaci ve 3d prostoru, například možnost intuitivního pohybu ve scéně pomocí klávesnice a myši.

2.2. TÉMA, TERÉN

Jako téma scény jsem zvolil zatopenou krajinu. Krajina nyní obsahuje stromy, křoví, trávu a vodu. Základem scény je „travnatý“ povrch, který se vlní napříč scénou. Tvoří se generováním pseudonáhodné výškové mapy, která se několikrát sčítá s rozdílným měřítkem. Tím se vytvoří drobné vlnění terénu i větší kopce a údolí.

2.3. TEXTURY

Jelikož textury nelze kvůli velikosti předpřipravit a načítat, veškeré textury se vyrábí procedurálně při startu aplikace. Textury jsou v projektu generovány na principu Perlinova šumu. Generování textury probíhá podobně jako generování terénu. Nejprve se vygeneruje jedna textura s náhodnými hodnotami a ta se poté v různém měřítku a s různými koeficienty sčítá do výsledné textury. Některé textury jsou předtím/potom vyhlazeny či jinak jednoduše upraveny, aby více odpovídaly jejich očekávanému vzhledu.

Pro rozmanitost některých objektů, například terénu, podporuje aplikace multitexturing. Například na terén se tedy nanáší nejen jedna textura trávy stále dokola, jako tapeta, ale pomocí multitexturingu se vloží i druhá textura, která terénu dává různé odstíny.

2.4. VODA, MRAKY

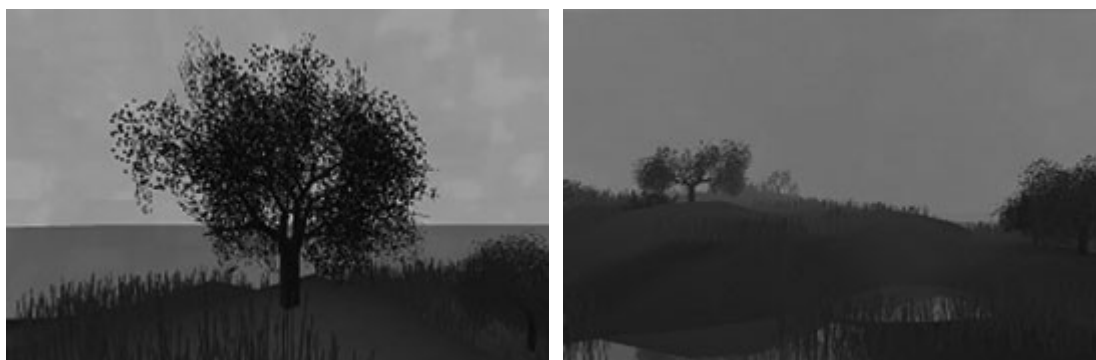
Pro věrnější obraz se ve vodní hladině zrcadlí krajina a všechny objekty nad vodou. Toho je docíleno pomocí jednoduchého zrcadlení celé scény pod vodní hladinu. Dalším detailem, který přidává na realističnosti je zobrazení mraků na principu skyboxu.

2.5. STROMY, KŘOVÍ, TRÁVA

Tvorba stromů a křoví se dělí na 2 části. Je třeba vygenerovat kmen a větve stromu a také je třeba vygenerovat shluky listí na jednotlivých větvích.

Generování větví stromu probíhá rekurzivně od kmene až k jednotlivým větvičkám dle předem stanovených pravidel pro daný typ stromu. Už při generování jednotlivých větví na určité úrovni se začnou generovat i shluky listí. Tyto shluky listí jsou jednoduché textury, obsahující náhodný počet jednotlivých lístků daného typu v daném barevném odstínu. Jednotlivé lístky jsou uspořádány tak, aby výsledná textura nevypadala hranatě, ale kulatě. Jakékoliv hrany nejsou v tomto případě žádoucí. U textur listí i trávy se využívá alfa-testing, díky kterému jsou nevyplněná místa v textuře průhledná. To umožňuje pomocí jedné textury vygenerovat velké množství malých lístků, které tedy není třeba zobrazovat jednotlivě.

Po vygenerování je tráva tvořena z mnoha vhodně uspořádaných textur. Tato textura trávy se vytváří jednoduchým kreslením stébel trávy pomocí Bresenhamova algoritmu.



Obrázek 1: Detail stromu, krajina

2.6. MLHA, VIDITELNOST, LOD

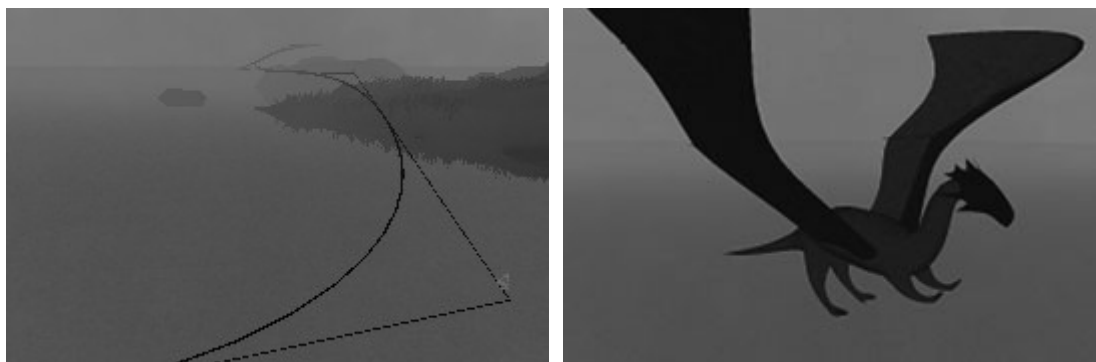
Kvůli rychlosti výsledné animace se zobrazuje ve scéně jen to, co je vidět. Ostatní objekty, které ve scéně nebudou zobrazeny se pomocí jednoduchého výpočtu předem vyřadí a nezatěžují zbytečně grafickou kartu.

Stromy obsahují různé úrovně detailu a pro vzdálenější objekty se nepoužije plnohodnotný model, ale jen zjednodušená verze, která je méně náročná na zobrazení. Bohužel těchto úrovní detailů je málo a při postupném vzdalování objektu jsou vidět znatelné skoky.

Pro větší realističnost a pro maskování některých druhů optimalizací scéna obsahuje mlhu. Ta je generována pomocí vestavěných algoritmů OpenGL, které stačí pouze zapnout.

2.7. POHYB KAMERY

Pro animaci je třeba, aby se pozorovatel automaticky pohyboval scénou po předem definované trase. Zároveň je třeba otáčení scény a synchronizace s časem. Pro pohyb a rotace po scéně se využívá NURBS křivek. Díky křivkám stačí v aplikaci zadat pouze několik kontrolních bodů, kolem kterých pozorovatel ve scéně v daném čase proletí. Tyto body se vyhladí do křivek a zajistí pěknou plynulost pohybu a rotací.



Obrázek 2: Křivka pro pohyb kamery, drak

2.8. DALŠÍ MODELY, SUBDIVISION

Pro další objekty ve scéně, které nelze jednoduše procedurálně vyrobit, je implementováno vložení polygonálních modelů vymodelovaných v některém grafickém programu. Tyto modely jsou předem překonvertovány na hlavičkové soubory a vloženy do programu jako sada proměnných.

Pro minimalizaci velikostí těchto modelů je využit algoritmus Catmull-Clark pro dělení povrchů. Pomocí tohoto algoritmu se dá z polygonálního modelu rekurzivně získat detailnější a hladší model, který by při normálním uložení zabíral řádově více.

2.9. ZVUK

Aplikace obsahuje zvukový doprovod. Pro minimalistické řešení nebylo možné použít například kvalitu mp3 souborů, ale sáhnout po méně paměťově náročném formátu ukládajícím víceméně jen seznam not melodie a k jejich interpretaci je použit externí syntetizátor.

3. ZÁVĚR

Díky vestavěným funkcím pro kreslení, křivky, textury a externí modely je možno zobrazit a animovat téměř libovolnou scénu při přijatelné velikosti výsledné aplikace.

LITERATURA

- [1] WWW stránky - CZ Nehe OpenGL, <http://nehe.ceske-hry.cz/>