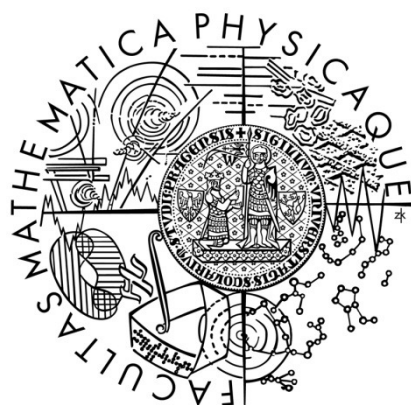


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Jiří Danihelka

Modelování terénu a jeho modifikací

Kabinet software a výuky informatiky

Vedoucí diplomové práce: Doc. Dr. Ing. Ivana Kolingerová

Studijní program: Informatika, Softwarové systémy

2008

Poděkování

Děkuji vedoucí diplomové práce za cenné připomínky k textu práce, které podstatným způsobem přispěly k jejímu zlepšení.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne

Obsah

| | |
|---|----|
| Abstrakt | 5 |
| 1. Úvod | 6 |
| 1.1. Dohodnuté zadání programu | 7 |
| 2. Existující nástroje pro 3D modelování..... | 8 |
| 2.1. Program 3D Studio Max..... | 8 |
| 2.2. Program Blender..... | 9 |
| 2.3. Virtuální světy pro běžné uživatele | 10 |
| 2.3.1. Herní virtuální světy | 10 |
| 2.3.2. Neherní virtuální světy | 11 |
| 3. Křivky v počítačové grafice | 12 |
| 3.1. Způsoby zápisu rovnice křivky | 12 |
| 3.1.1. Explicitní zápis rovnice křivky | 13 |
| 3.1.2. Parametrický zápis rovnice křivky | 14 |
| 3.1.3. Implicitní zápis rovnice křivky | 15 |
| 3.1.4. Porovnání jednotlivých způsobů reprezentace křivek..... | 15 |
| 3.2. Navazování a spojitost křivek | 16 |
| 3.2.1. Parametrická spojitost | 16 |
| 3.2.2. Třídy geometrické spojitosti..... | 17 |
| 3.2.3. Rozdíl mezi parametrickou a geometrickou spojitostí..... | 17 |
| 4. Zvolené metody editace terénů | 18 |
| 4.1. Volba plátů použitých při modelování..... | 18 |
| 4.2. Výběr plátů | 19 |
| 5. Bézierovy pláty v počítačové grafice | 20 |
| 5.1. Obecné Bézierovy pláty | 20 |
| 5.2. Bikubické Bézierovy pláty | 20 |
| 5.3. Princip převodu na síť trojúhelníků | 20 |
| 5.4. Vyhledání průsečíku Bézierova plátu s paprskem | 21 |
| 6. Texturování terénu..... | 23 |
| 6.1. Obecné způsoby texturování..... | 23 |
| 6.2. Automatické generování souřadnic textury | 23 |

| | | |
|--------|---|----|
| 6.3. | Rozdíl mezi spojitostí C1 a G1 při texturování..... | 24 |
| 7. | Výběr grafické knihovny | 25 |
| 7.1. | Knihovna OpenGL | 25 |
| 7.2. | Knihovna DirectX | 26 |
| 7.3. | Knihovna Java3D | 27 |
| 7.3.1. | Kritika knihovny Java3D..... | 27 |
| 8. | Vytvořený program Editor terénů | 30 |
| 8.1. | Obecný popis programu | 30 |
| 8.2. | Instalace a spuštění Editoru terénů..... | 31 |
| 8.2.1. | Platformy Windows XP 32 bit, Windows Vista 32 bit | 31 |
| 8.2.2. | Ostatní platformy Windows | 31 |
| 8.2.3. | Ostatní platformy – Linux, Solaris, Mac Os | 31 |
| 9. | Ovládání Editoru terénů | 32 |
| 9.1. | Hlavní okno Editoru terénů | 32 |
| 9.2. | Ovládání kamery v Editoru terénů..... | 33 |
| 9.2.1. | Ovládání kamery pomocí myši | 33 |
| 9.2.2. | Ovládání kamery pomocí klávesnice | 34 |
| 9.3. | Manipulace s Bézierovými pláty | 34 |
| 9.3.1. | Sdílení řídicích bodů mezi Bézierovými pláty..... | 35 |
| 9.3.2. | Manipulace s řídicími body Bézierových plátů..... | 36 |
| 9.3.3. | Popis ovládacích prvků záložky Terrain..... | 37 |
| 9.3.4. | Popis ovládacích prvků záložky Tools..... | 39 |
| 9.3.5. | Popis ovládacích prvků záložky Show/Hide | 41 |
| 9.3.6. | Popis ovládacích prvků záložky Mass move | 42 |
| 10. | Závěr | 45 |
| 10.1. | Vyhodnocení splnění zadání programu | 45 |
| 10.2. | Ukázky vymodelovaných terénů | 47 |
| | Seznam rovnic | 49 |
| | Seznam obrázků | 50 |
| | Použitá literatura..... | 51 |
| | Obsah přiloženého CD | 53 |

Abstrakt

Název práce: Modelování terénu a jeho modifikací

Autor: Jiří Danihelka

Katedra: Katedra softwarového inženýrství

Vedoucí diplomové práce: Doc. Dr. Ing. Ivana Kolingerová, ZČU Plzeň

E-mail vedoucího: kolinger@kiv.zcu.cz

Abstrakt: Diplomová práce se zabývá možnostmi intuitivního modelování terénu pro potřeby počítačových her. Hledá se taková metoda, která by umožnila snadné modelování i uživateli, který nemá zkušenosti s vytvářením 3D objektů, a zároveň poskytovala široké možnosti, aby se dala použít i odborníky na tvorbu počítačových her. V práci se diskutuje nad vhodnou volbou metody modelování a různé metody jsou vzájemně srovnávány. Jako nejvhodnější je vybrána metoda modelování pomocí přesouvání řídicích bodů u Béziových plátů, které se práce pak blíže věnuje. V diplomové práci je navržen a představen program Editor terénů, který navrženou metodu implementuje a demonstruje v praxi její použití.

Klíčová slova: modelování, terén, plát, Bézier

Title: Modeling of terrain and its modifications

Author: Jiří Danihelka

Department: Department of Software Engineering

Supervisor: Doc. Dr. Ing. Ivana Kolingerová, ZČU Plzeň

Supervisor's e-mail address: kolinger@kiv.zcu.cz

Abstract:

This thesis is about intuitive terrain modeling for needs of computer games. The thesis is searching for a method of terrain modeling that will be easy for people with lack of experience. The method also should be able to allow powerful modeling techniques for professionals game creators. The thesis debate about choosing appropriate method and compare different methods between each other. Selected is the method of moving control points of Bezier patches. The thesis presents a program Terrain Editor that was created as part of the thesis. The program implements selected method and demonstrates its use in practice.

Keywords: modeling, terrain, patch, Bezier

1. Úvod

V dnešní době se stále více setkáváme s pronikáním trojrozměrné grafiky do světa osobních počítačů. V některých oblastech se však přechod od dvourozměrného k trojrozměrnému potýká s obtížemi. Například webové stránky jsou v naprosté většině případů stále pouze dvojrozměrné. Jedním z hlavních příčin těchto problémů je náročné vytváření trojrozměrného obsahu virtuální světů. Běžní lidé totiž, kteří vytvářejí webové stránky, mají zájem vytvářet trojrozměrný obsah, v současné době je však proces tvorby tak náročný, že od něho většinou upouštějí. Pokud by byly vytvořeny vhodné metody modelování trojrozměrných objektů, použití prostorové grafiky by se masově rozšířilo i mimo oblast profesionálních modelářů. K tomuto cíli chce přispět i tato diplomová práce, která se zabývá možnostmi snadného modelování terénů.

Cílem diplomové práce je navrhnout metodu modelování terénu pro počítačové hry, která by byla snadno pochopitelná i pro běžného uživatele, který nemá s trojrozměrným modelováním žádné zkušenosti. Zároveň je potřeba, aby navržená metoda nabízela takové možnosti editování, aby ji mohli používat profesionální modeláři při tvorbě her.

Aby bylo možné navrženou metodu otestovat, zda je opravdu vhodná pro praktické modelování, v rámci diplomové práce byl implementován editor, který umožňuje provádět modifikace terénu pomocí navržené metody.

Zdroj: (1), (2)

1.1. Dohodnuté zadání programu

Požadavky na software, který vzniknul v rámci diplomové práce, jsem stanovil podle požadavků pana inženýra Michala Varnušky PhD. Pan Varnuška je vývojářem počítačových her ve společnosti Black Element Software (<http://www.blackelement.net/>). Zabývá se také programováním nástrojů pro návrháře a designéry počítačových her. Díky jeho zpětné vazbě a zkušenostem z modelování byla zlepšena kvalita editoru.

Během počáteční fáze vývoje bylo dohodnuto následující zadání na tvorbu editoru:

Požadavky na produkt

- Práce s editorem bude interaktivní a bude probíhat v reálném čase.
- Při vytváření nového terénu uživatel zvolí rozměry terénu v km a počet plátů, na které bude terén rozdělen.
- Editor bude umět exportovat data v podobě výškové mapy v rozlišení $32 \times x + 1$ a normálové mapy v rozlišení $32 \times x \times \text{SIZE} + 1$, kde x je libovolné přirozené číslo zadané uživatelem. Editor také bude umět exportovat souřadnice pro texturování.
- Editace plátů bude probíhat změnou polohy řídicích bodů plátů v prostoru. Při návrhu GUI, bude kladen důraz na jednoduchost editování terénu.
- Uživatel bude moci nastavit stupeň spojitosti mezi pláty na hodnoty G^0 nebo G^1

Požadavky na proces

- Vhodný programovací jazyk a grafické knihovny pro práci si zvolí řešitel. Zadavatel v tomto ohledu nemá požadavky. Jazyky .NET nejsou preferované.
- Řešitel se podrobně seznámí s dostupnými křivkami a plochami a vybere vhodné řešení pro použití v editoru.

Systémové požadavky pro spuštění editoru

- Windows XP nebo Windows Vista
- grafická karta s podporou shaderů
- Před spuštěním editoru může být požadována instalace nebo upgrade volně dostupných grafických nebo jiných knihoven třetí stran. (např.: DirectX, JRE, Ogre, Java3D)

2. Existující nástroje pro 3D modelování

V současnosti je dostupných mnoho nástrojů pro obecné trojrozměrné modelování. Tyto nástroje jsou často primárně určeny k vytváření modelů pro počítačové hry (modelů pro real-time počítačovou grafiku), pro renderování videa (non real-time) nebo k obojímu.

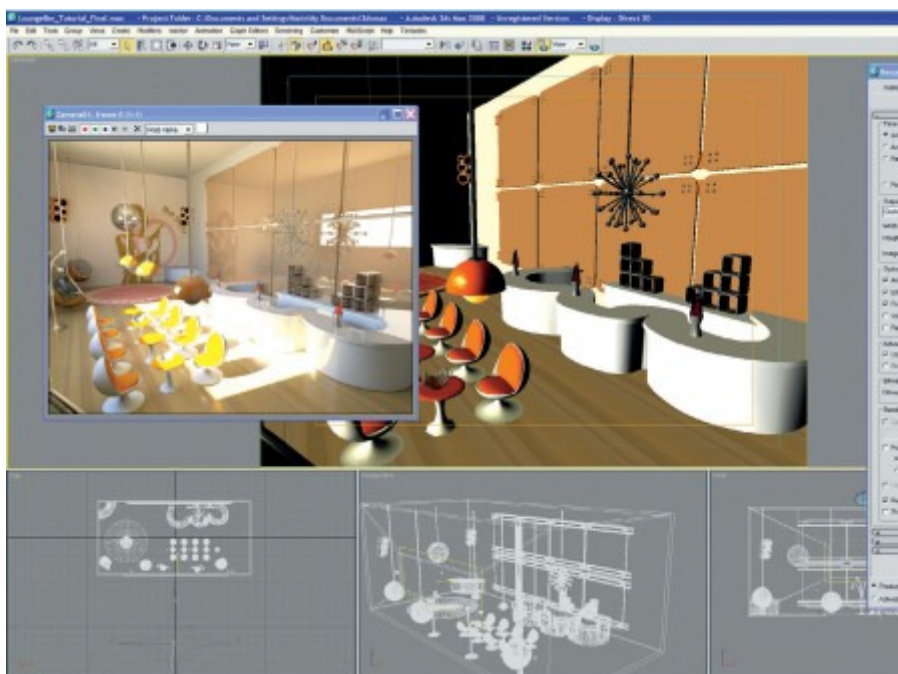
Uvedu zde stručný popis dvou nejznámějších nástrojů pro trojrozměrné modelování: 3D Studio Max a Blender

2.1. Program 3D Studio Max

Program 3D Studio Max (Obrázek 1) je celosvětově nejprodávanější 3D modelovací, animační a renderovací software používaný při postprodukci filmů, výrobě reklam, pro architektonické vizualizace a tvorbu modelů do počítačových her. Používá ho 80% všech herních vývojářů.

Možnosti programu 3D Studio Max jsou opravdu rozsáhlé. Program obsahuje několik technologií pro rendering (ray-tracing, radiosity, global illumination), používá vlastní skriptovací jazyk MaxScript a podporuje dlouhou řadu souborových formátů pro 3D grafiku. Nové verze umějí i simulovat fyzikální srážky pevných i měkkých těles. Další funkčnost lze do 3D Studio Max přidávat pomocí zásuvných modulů.

O vývoj programu 3D Studio Max se v současnosti stará firma Autodesk. Program 3D Studio Max je určen výlučně pro profesionální modeláře. Tomu odpovídá obtížnější způsob ovládání i cena za pořízení, která převyšuje 140 000 Kč.



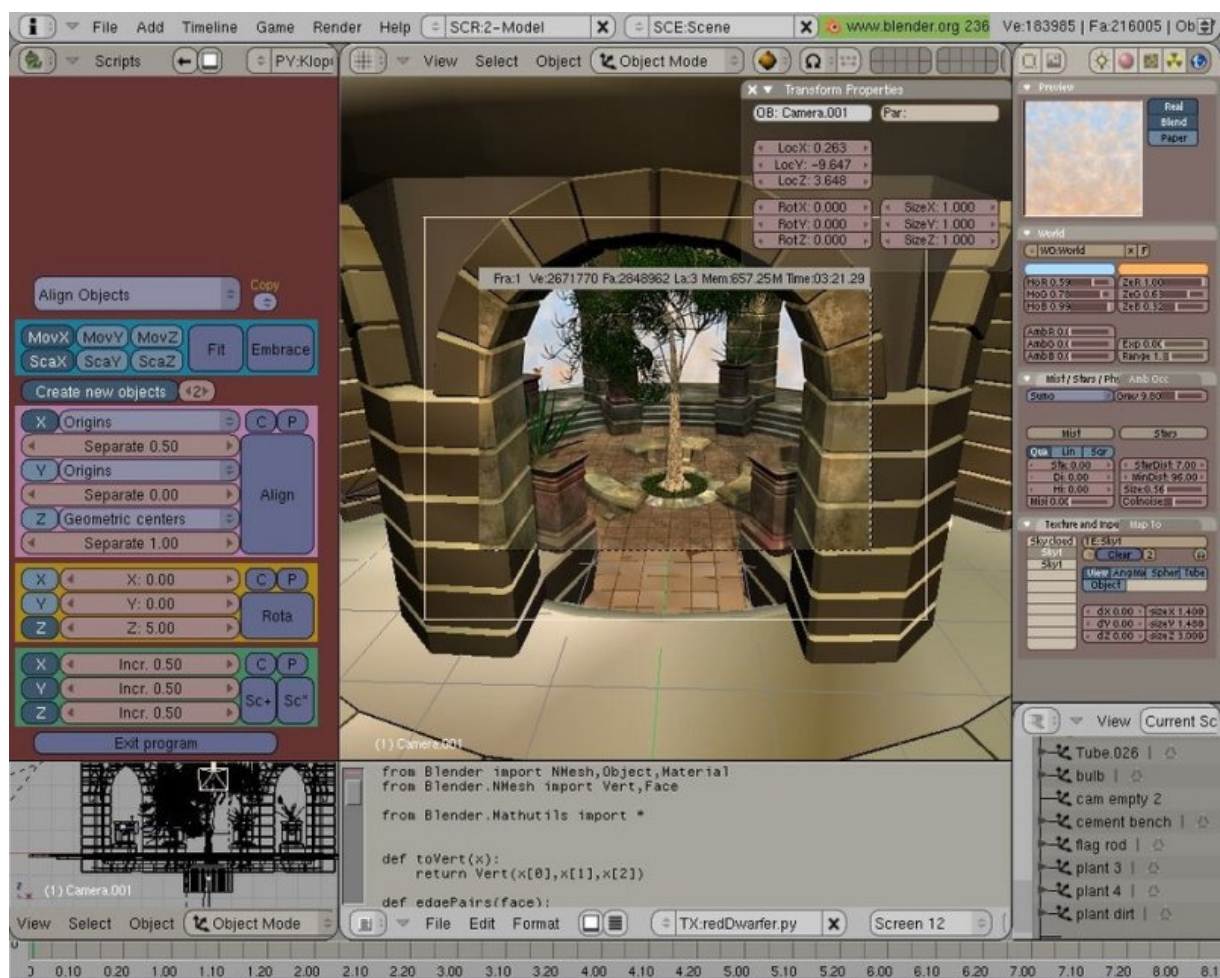
Obrázek 1: Nástroj pro 3D modelování 3D Studio Max

Zdroj: (3), (4)

2.2. Program Blender

Program Blender (Obrázek 2) je open source alternativou k programu 3D Studio Max. Populární se stal hlavně pro uživatele operačního systému Linux. Používá se pro vykreslování počítačové grafiky a tvorbě animací pomocí metod sledování paprsku a radiosity. Celý program je založen na grafické knihovně OpenGL a mezi jeho velké přednosti patří snadná přenositelnost mezi platformami. Mezi jeho nevýhody patří komplikovaný způsob ovládání, kvůli kterému je program schopen dobře používat jen profesionální modelář nebo člověk velmi zkušený v modelování.

Původně byl program Blender vyvíjen holandskou firmou Not A Number jako profesionální software pro modelování. Když roku 2002 firma Not A Number zkrachovala, podařilo se pomocí sponzorských darů licenci vykoupit a Blender se stal svobodným softwarem. Nyní je Blender distribuován pod licencí GNU GPL a o jeho vývoj se stará Blender Foundation.



Obrázek 2: Nástroj pro 3D modelování Blender

Zdroj: (5), (6)

2.3. Virtuální světy pro běžné uživatele

Profesionální nástroje pro tvorbu trojrozměrných modelů jsou pro běžného uživatele příliš komplikované. Existují však jiné možnosti. Vzniklo několik virtuálních světů, do kterých může běžný uživatel vstoupit a někdy se i podílet se na vytváření jeho obsahu. Virtuální světy se rozdělují na herní (gaming virtual worlds) a neherní (non-gaming virtual worlds).

2.3.1. Herní virtuální světy

Herní virtuální světy pro Internetové hry obvykle umožňují pouze omezené vytváření obsahu jeho uživateli. Tvůrci her chtějí mít jistotu, že obsah hry bude kvalitní a proto příliš nedovolují uživatelům do virtuálního světa zasahovat změnou obsahu. Typickým zástupcem herního virtuálního světa může být počítačová hra World of Warcraft (Obrázek 3).

Počítačové hry občas obsahují i editor map, pomocí kterého si uživatel může vytvořit vlastní úroveň hry. Výrobce hry někdy nabízí možnost, aby mu hráči povedené mapy poslali. Výrobce hry pak vybere několik nejlepších map a zveřejní je na webové stránce hry.



Obrázek 3: Počítačová hra World of Warcraft

Zdroj: (7)

2.3.2. Neherní virtuální světy

V neherních virtuálních světech nejsou uživatelům dány žádné cíle nebo hrdinské úkoly, které mají plnit. Uživatelé chodí do neherních virtuálních světů chatovat s ostatními obyvateli, prozkoumávat lokace a jen tak se bavit.

Tvůrci neherních virtuálních světů dávají uživatelům široké možnosti, jak se mohou podílet na obsahu virtuálního světa. Například v nejznámějším neherním virtuálním světě Second Life (Obrázek 4) od společnosti Linden Lab si mohou už uživatelé navrhnout a postavit vlastní virtuální dům a vybavit ho virtuálním nábytkem. Vytváření virtuálního obsahu je pro běžného uživatele srozumitelné, ale nenabízí příliš široké možnosti. Často se jedná jen o pouhé vybírání předmětů z katalogu. Pokud chce uživatel vytvářet vlastní objekty a třeba pro ně nahrát textury na server Second Life, musí provozovatelům serveru zaplatit. Uživatel nemá možnost modelovat terén v Second Life, ten modelují pouze lidé z Linden Lab.

Alternativou k virtuálnímu světu Second Life je neherní virtuální svět od společnosti Google Lively (Obrázek 5). Zatím je však projekt spuštěn jen krátce a k dispozici je jenom jeho beta verze.



Obrázek 4: Virtuální svět Second Life



Obrázek 5: Virtuální svět Lively

Zdroj: (8)

3. Křivky v počítačové grafice

V počítačové grafice se pro modelování používají křivky a plochy. Existuje mnoho druhů křivek a ploch, ne všechny jsou však vhodné k intuitivnímu modelování terénu. Různé přístupy k modelování pomocí křivek mezi sebou porovnáme.

3.1. Způsoby zápisu rovnice křivky

Existuje několik způsobů, jak zapsat a reprezentovat křivku v počítačové grafice. Pokusím se jednotlivé metody porovnat z hlediska vhodnosti pro aplikaci modelování terénů. Nejdůležitější požadavkem při výběru reprezentace křivky je to, zda je možné ji snadno převést do tvaru vhodného pro vykreslení.

Častým požadavkem při používání křivek v počítačové grafice a počítačových hrách také bývá určení tečného vektoru. Pomocí tečného vektoru se určí normálový vektor. Ten je při zobrazování herní scény velmi důležitý jelikož se používá ve všech netriviálních osvětlovacích modelech. Normálový vektor má zásadní vliv na barvu, kterou počítač vykreslí virtuální objekt na obrazovku.

Pokud jsme v dvourozměrném případě, normálový vektor z tečného vypočteme snadno. Mějme tečný vektor $t = (x, y)$, pak normálové vektory mají tvar $n_1 = (-y, x)$ a $n_2 = (y, -x)$.

V případě trojrozměrného prostoru existuje kolmých vektorů ke křivce v kterémkoliv bodě nekonečně mnoho. Pokud chceme určit normálový vektor nějakého povrchu v určeném bodě, musíme znát nejméně 2 křivky, které určeným bodem procházejí. Dále zjistíme tečné vektory obou křivek v určeném bodě. Normálový vektor pak zjistíme jako normalizovaný vektorový součin obou tečných vektorů (viz Rovnice 1).

$$n = \frac{t_1 \times t_2}{\|t_1 \times t_2\|}$$

Rovnice 1: Výpočet normálového vektoru v trojrozměrném případě

Zdroj: (9 stránky 177-236), (10)

3.1.1. Explicitní zápis rovnice křivky

Explicitním zápisem rovnice křivky rozumíme takový zápis, ve kterém je jedna ze souřadnic (obvykle x) nezávislým parametrem v určitém intervalu a ostatní souřadnice (y – ve dvourozměrném případě; y, z – ve trojrozměrném případě) na ní funkčně závisí. Explicitní zápis rovnice křivky má několik nevýhod. Takovýmto způsobem lze zapsat pouze křivky, které jsou zároveň funkcemi, jelikož k jedné hodnotě x nelze přiřadit více hodnot y . S funkcí v explicitním zápisu se špatně manipuluje, komplikovaná je například operace otočení.

Nevýhody explicitního zápisu se dají částečně obejít. Komplikovanější tvary křivek, které se nedají zapsat, jako matematická funkce, je možné zapsat jako sjednocení několika křivek v explicitním zápisu. Například pokud chceme rovnici kružnice (viz Rovnice 2) popsat explicitně, nevystačíme si s jedinou rovnicí a musíme kružnici popsat jako sjednocení minimálně dvou explicitně zapsaných křivek.

$$\begin{aligned}y_1 &= -\sqrt{R^2 - x^2} \\y_2 &= +\sqrt{R^2 - x^2} \\x &\in \langle -R, R \rangle\end{aligned}$$

Rovnice 2: Explicitní vyjádření rovnice kružnice se středem v počátku a poloměrem R

Křivka popsaná explicitním způsobem je spojitá právě tehdy, když je spojitá funkce, podle které se vypočítají závislé souřadnice. Tečný vektor k funkci v explicitním tvaru vypočítáme pro dvojrozměrný případ podle rovnice 3.

$$t = \left(1, \frac{\partial y}{\partial x}\right)$$

Rovnice 3: Výpočet tečného vektoru z explicitního tvaru rovnice křivky ve 2D případě

V trojrozměrném případě bude způsob výpočtu obdobný (viz Rovnice 4).

$$t = \left(1, \frac{\partial y}{\partial x}, \frac{\partial z}{\partial x}\right)$$

Rovnice 4: Výpočet tečného vektoru z explicitního tvaru rovnice křivky ve 3D případě

Zdroj: (9 stránky 177-236)

3.1.2. Parametrický zápis rovnice křivky

Parametrická křivka používá oproti ostatním způsobům zápisu nový parametr, na kterém jsou ostatní souřadnice explicitně závislé. Parametr nabývá hodnot z předem stanoveného intervalu, kterým je nejčastěji interval $\langle 0,1 \rangle$. Rovnici v parametrickém tvaru si také můžeme představit jako trajektorii bodového tělesa, kde parametr má význam času.

Příkladem parametrické rovnice je rovnice kružnice v parametrickém zápisu. (viz Rovnice 5).

$$x = R \cdot \cos(u \cdot 2\pi)$$

$$y = R \cdot \sin(u \cdot 2\pi)$$

$$u \in \langle 0,1 \rangle$$

Rovnice 5: Parametrické vyjádření rovnice kružnice se středem v počátku a poloměrem R

Tečný vektor je možné vypočítat ve dvojrozměrném případě z rovnice 6 a v trojrozměrném případě z rovnice 7. V obou případech není výpočet nijak zvlášť náročný.

$$t = \left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u} \right)$$

Rovnice 6: Výpočet tečny ve 2D případě

$$t = \left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u} \right)$$

Rovnice 7: Výpočet tečny ve 3D případě

Zdroj: (9 stránky 177-236)

3.1.3. Implicitní zápis rovnice křivky

Rovnice křivky zapsaná v implicitním tvaru je zapsaná pomocí rovnosti závislé na proměnných x a y . Tam, kde je rovnost splněna, se nachází body náležející křivce. Rovnice v implicitním tvaru nemusí být (a obvykle ani není) rozřešena vzhledem k žádné ze tří proměnných. Pokud přejdeme z dvojrozměrného do třírozměrného prostoru, implicitní křivky přejdou na implicitní plochy. Přestože implicitní plochy mají počítačové grafice také svá využití, implicitní křivky se v třídimenzionálním prostoru rozumně používat nedají.

Zápis křivky implicitním způsobem může být jednodušší (viz Rovnice 8) než při zápisu explicitním. Někdy je možné převést implicitní zápis křivky na explicitní alespoň pro část křivky. Často to však bývá velmi náročné.

$$x^2 + y^2 = R^2$$

Rovnice 8: Implicitní vyjádření rovnice kružnice se středem v počátku a poloměrem R

Pokud máme rovnici implicitní funkce zapsanou ve tvaru $F(x, y) = 0$, tečný vektor určíme pomocí rovnice 9.

$$t = \left(1, -\frac{\frac{\partial F}{\partial x}}{\frac{\partial F}{\partial y}}\right)$$

Rovnice 9: Výpočet tečného vektoru z implicitního tvaru rovnice křivky ve 2D případě

Zdroj: (9 stránky 177-236)

3.1.4. Porovnání jednotlivých způsobů reprezentace křivek

Každé ze tří způsobů vyjádření rovnice křivky má své výhody a nevýhody. Křivka zapsaná v explicitním tvaru se snadno sice vykresluje, ale má vážné problémy při provádění lineárních transformací, obzvláště pak rotací. S křivkou v implicitním zápisu lze snadno provádět lineární i nelineární transformace, ale často je obtížné zjistit, jaké body křivce náleží, a tudíž bývá problematické přímku vykreslit. Jako nejlepší možnost vychází používat parametrický zápis rovnice křivky, jelikož umožňuje snadné provádění lineárních i nelineárních transformací a též je možné křivku snadno vykreslit.

Dále budeme uvažovat, že křivky a plochy, se kterými pracujeme, jsou vyjádřené v parametrickém tvaru, jelikož tento tvar je pro počítačovou grafiku nejvýhodnější.

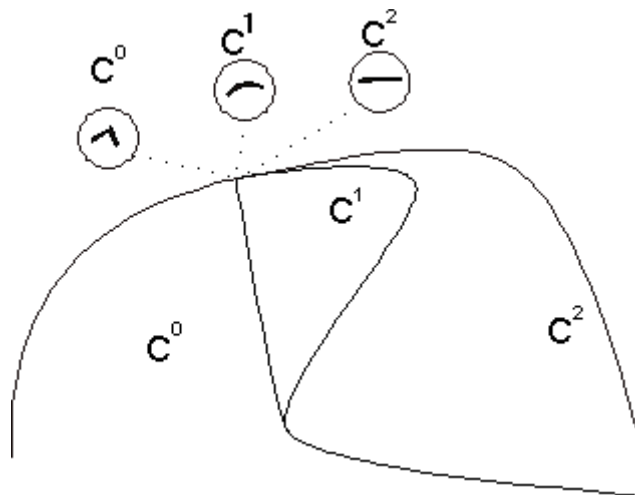
3.2. Navazování a spojitost křivek

V praxi je často potřeba, aby na sebe křivky hladce navazovaly. V počítačové grafice jsou pro vyjádření míry hladkosti navázání křivek zavedeny stupně spojitosti. Pro každé spojení dvou křivek se dá určit, do jakého stupně spojitosti patří. Pokud máme nějaké požadavky na hladkou spojitost křivek, můžeme tyto požadavky často vyjádřit pomocí stupňů spojitosti.

3.2.1. Parametrická spojitost

Stupně parametrické spojitosti se označují $C^0, C^1, C^2, C^3, \dots$. Spojení dvou křivek je stupně C^0 , právě když koncový bod první křivky je zároveň počátečním bodem druhé křivky. Spojení dvou křivek je stupně C^1 , pokud je stupně C^0 a zároveň derivace první křivky v koncovém bodě je rovna derivaci druhé křivky v prvním bodě. Geometrická představa této podmínky je taková, že se při průchodu bodem spojení dvou křivek zachovává směr i velikost rychlosti. Rozdíl mezi stupni parametrické spojitosti je zobrazen na obrázku 6.

Spojení dvou křivek je stupně C^n právě tehdy kdy jsou všechny parciální derivace až do řádu n všech souřadnic v posledním bodě první křivky rovny parciálním derivacím řádu n v prvním bodě druhé křivky. Pokud je spojení dvou křivek stupně C^n , pak je i všech nižších stupňů. Opačná implikace obecně neplatí.



Obrázek 6: Vizuální rozdíl mezi parametrickými třídami spojitosti

Zdroj: (11)

3.2.2. Třídy geometrické spojitosti

Třídy geometrické spojitosti se označují $G^0, G^1, G^2, G^3, \dots$. Spojení dvou křivek je ve třídě G^0 za stejných podmínek, které platí pro třídu C^0 , tedy koncový bod první křivky musí být shodný s počátečním bodem druhé křivky. Spojení křivek je ve třídě G^n právě tehdy, když je ve třídě G^{n-1} a zároveň existuje kladná konstanta h tak, aby platila rovnice 10, ve které t_1 a t_2 jsou parametry rovnic spojovaných křivek. Parciální derivace se porovnávají v místě, kde dochází ke spojení obou křivek.

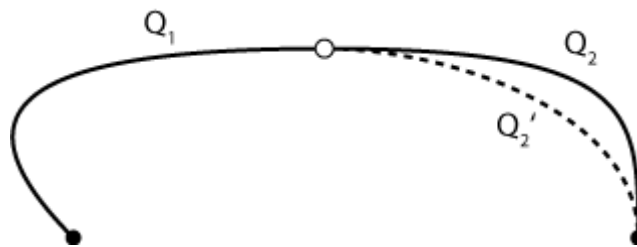
$$\left[\frac{\partial^n x}{\partial t_1^n}, \frac{\partial^n y}{\partial t_1^n}, \frac{\partial^n z}{\partial t_1^n} \right] = h \cdot \left[\frac{\partial^n x}{\partial t_2^n}, \frac{\partial^n y}{\partial t_2^n}, \frac{\partial^n z}{\partial t_2^n} \right]$$

Rovnice 10: Podmínka geometrické spojitosti řádu n

3.2.3. Rozdíl mezi parametrickou a geometrickou spojitostí

V počítačové grafice se běžně setkáváme se spojitostmi C^0, C^1, C^2 a G^1 . S ostatními druhy spojitostí se v praxi setkáme jen zřídka. Pokud jsou křivky nakreslené, bývá těžké rozpoznat, zda je jejich spojitost C^1 nebo G^1 . Pokud se však jedná o animační křivku, po které se pohybuje nějaký objekt, rozdíl ve spojitostech už bude patrný. Spojitost C^1 zaručuje, že v bodě návaznosti křivek bude stejná poloha a rychlost objektu. Spojitost G^1 zaručuje pouze stejnou polohu a to, že rychlost bude mít stejný směr, ale může se změnit skokovitě. Při animaci je skokovitá změna rychlosti patrná, a proto je vhodnější použít spojitost třídy C^1 .

Rozdíl mezi parametrickou a geometrickou spojitostí je zobrazen na obrázku 7. Horní křivka navazuje se spojitostí C^1 , dolní navazuje se spojitostí G^1 .



Obrázek 7: Parametrická a geometrická spojitost.

Zdroj: (9), (12)

4. Zvolené metody editace terénů

4.1. Volba plátů použitých při modelování

Při volbě plátů, které se budou používat v editoru při modelování terénu, bylo usilováno, aby editace byla pro uživatele co nejvíce intuitivní. Nejdříve byl proto implementován obecný editor, který umožňoval editaci pomocí několika druhů plátů a křivek a následně byl vybrán způsob, který se zdál být nejlepší.

Porovnáním editace pomocí různých plátů jsem zjistil několik poznatků:

- a) Editování objektů pomocí přesouvání řídicích bodů je pro uživatele mnohem snazší než zadávání parametrů textově. Pokud má uživatel vymodelovat přímku požadovaného tvaru a směru dosáhne mnohem snadněji pomocí umístění dvou bodů do prostoru, kterými se přímka proloží, než kdyby měl napsat koeficienty v rovnici přímky. Tím nechci říci, že zadávání koeficientů je obecně špatná metoda. Záleží na způsobu, kterým chceme modelovat. Pokud modelujeme průběhy funkcí v nějaké matematické aplikaci, je zadávání parametrů funkce vhodnější než přesouvání řídicích bodů. Pro metody tvůrčího modelování, do kterého počítám i editaci terénů, je však mnohem lepší používat metodu přesouvání řídicích bodů. Rozhodl jsem se proto zaměřit se na metody, které budou založeny na přesouvání řídicích bodů.
- b) Zadávání číselných parametrů v řídicích bodech je pro uživatele také nepřirozené. Některé křivky používají pro modelování kromě řídicích bodů také parametry, u kterých je potřeba nastavit hodnoty v řídicích bodech. Nejčasněji se jedná o parametr váhy v řídicího bodu. Některé zobecněné křivky jako NURBS nebo racionální B-spline používají pro modelování zároveň řídicí body a parametr váhy bodu. Vkládání takovýchto parametrů je však značně neintuitivní a editování komplikuje. Proto jsem se rozhodl použít metodu, která nebude vyžadovat zadávání dodatečných parametrů.
- c) Pro uživatele je velmi problematické modelovat objekty, ve kterých nejsou žádné interpolační body. Například Coonsovy kubiky nemají žádné interpolační body. Všechny řídicí body Coonsových kubik jsou aproximační, a proto se špatně modelují. Čím více interpolačních bodů má uživatel k dispozici při modelování terénu, tím je pro něho modelování snazší.

d) Pro uživatele je velmi nepřírozené zadávat v řídicích bodech tečné nebo normálové vektory. Takový způsob modelování je velmi neintuitivní. Některé křivky a pláty pro modelování potřebují zadat kromě řídicích bodů tečné nebo normálové vektory v řídicích bodech. Jedná se vždy o vektory interpolační, protože v aproximačních bodech by nemělo stanovování tečných ani normálových vektorů smysl. V případě tečných vektorů se u křivek zadává jeden tečný vektor a u plátů dva tečné vektory. Normálové vektory se zadávají pouze u plátů, protože na křivku lze v libovolném bodě umístit nekonečně mnoho vektorů, které na ni budou v daném místě kolmé. Například Hermitovská kubika vyžaduje při modelování stanovení tečných vektorů ve svých krajních bodech. Modelování Bézierových křivek nevyžaduje stanovení tečných vektorů, a proto je pro uživatele snazší než práce s Hermitovskými křivkami. Bézierovy křivky se také v praxi více používají než křivky Hermitovské.

Poznámka:

To, zda je při použití nějaké editační metody potřeba zadávat nějaké textové parametry, záleží nejen na dané metodě, ale i na implementaci uživatelského rozhraní při použití dané metody. Pokud v textu uvádím, že nějaká metoda potřebuje zadávání dodatečných parametrů, znamená to, že neznám žádný přirozený způsob, jak se zadávání parametrů textově v takovémto případě vyhnout. Lze si však představit metody editace, které by textové zadávání nepoužívaly. Pokud například ke každému řídicímu bodu je potřeba nastavit tři parametry z oboru reálných čísel, uživatel by je mohl nastavit vybráním řídicího bodu pomocí myši, potom by zmáčknul jednu ze tří předurčených kláves podle parametru, který chce měnit, a pomocí pohybu myši by parametr zvětšoval nebo zmenšoval. Takovýto způsob lze použít vždy obecně, ale já jsem se rozhodl nic takového nepoužívat.

4.2. Výběr plátů

Nakonec jsem jako hlavní metodu pro editování terénu vybral Bézierovy pláty. Modelování pomocí Bézierových plátů splňuje všechny stanovené požadavky pro intuitivní modelování terénu. Bézierovy pláty se editují pomocí přesouvání řídicích bodů, není potřeba zadávat žádné další parametry. Z Bézierových plátů se jako nejvhodnější pro editaci jeví bikubické Bézierovy pláty, protože obsahují hodně interpolačních bodů, které jsou pro uživatele snadno pochopitelné, a dobře se s nimi pracuje.

5. Bézierovy pláty v počítačové grafice

5.1. Obecné Bézierovy pláty

Výhodou Bézierových plátů je jednoduché modelování, snadná diferencovatelnost a možnost relativně snadno spočítat průsečík plátu s paprskem.

Obecný Bézierův plát $n \times m$ -tého stupně je stanoven dvojrozměrným polem řídicích bodů o rozměrech $(n + 1) \times (m + 1)$. Řídící body z tohoto pole označujeme $P_{i,j}$, kde i je řádek pole a j je sloupec pole.

Rovnice obecné Bézierova plátu je $Q(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} B_i^n(u) B_j^m(v)$

kde

v je parametr plochy z intervalu $\langle 0, 1 \rangle$

u je parametr plochy z intervalu $\langle 0, 1 \rangle$

$P_{i,j}$ jsou řídicí bod Bézierova plátu

$B_i^n(u)$ je Bernsteinův polynom n -tého stupně $B_i^n(u) = \binom{n}{i} u^i (1 - u)^{n-i}$

5.2. Bikubické Bézierovy pláty

Bikubický Bézierův plát je speciálním případem obecného Bézierova plátu, pro který platí $n = m = 3$.

Odtud plyne název bikubický. Jelikož obecný Bézierův plát je určen $(n + 1) \times (m + 1)$ řídicími body, bude bikubický Bézierův je určen 4×4 řídicími body, tedy šestnácti řídicími body.

5.3. Princip převodu na síť trojúhelníků

Často požadovanou operací při práci s plochami je převést plochu do podoby, kterou lze rychle a snadno vykreslit na počítači. Obecný tvar plochy se často pro real-time vykreslování nehodí a dá se použít jen při off-line vykreslování pomocí metody ray-tracingu. Současné grafické karty počítačů jsou optimalizovány na rychlé vykreslování trojúhelníků a jiných polygonů. Abychom byli schopni v reálném čase na počítači vykreslovat nějakou plochu, je potřeba ji převést na síť polygonů.

Zdroj: (9)

5.4. Vyhledání průsečíku Bézierova plátu s paprskem

Při práci s pláty bývá potřeba umět vypočítat průsečík plátu s paprskem. Uvedu základní případy, kdy je výpočet průsečíku potřeba:

1. Uživatel bude vybírat pomocí myši pláty, se kterými bude provádět další operace
Pokud uživatel zadá pomocí myši příkaz k vybrání plátu, je potřeba zjistit, na který plát kurzor myši ukazuje. Pokud povedeme polopřímku (paprsek) z kamery do místa, kde se nachází kurzor myši, celá tato polopřímka bude ležet v jediném pixelu, na který ukazuje kurzor myši. Na této polopřímce leží body, které se při transformaci z 3D prostoru na 2D obrazovku monitoru zobrazí v daném pixelu. Pokud spočítáme průsečík této polopřímky s nějakým objektem, získáme bod v 3D prostoru, na který ukazuje kurzor myši. Nalezením průsečíku úloha nalezení plátu, na který ukazuje uživatel, nekončí. Je potřeba najít průsečíky se všemi pláty a porovnat vzdálenosti jejich průsečíků od kamery. Uživatel pomocí myši vybral ten plát, který je kameře nejbližší.

Výše uvedený postup není jediný, jak by šlo vybrat objekt pomocí myši. Teoreticky by bylo možné vyhnout se výpočtu průsečíků, pokud bychom například při vykreslování plátu ukládali do Z-bufferu kromě vzdálenosti i informaci, o který plát se jedná. Potom Takovýto postup by byl více paměťově náročný a jeho implementace by byla obtížnější.

2. Je potřeba exportovat výškovou mapu terénu
Vymodelovaný terén je potřeba převést do formátu, který je možné načíst v počítačové hře nebo v jiném editoru. Procesu, ve kterém se převádí soubor z nativního formátu editoru do formátu určeného pro jiné využití, se říká exportace. Množství počítačových her pracuje s výškovou mapou terénu. Reprezentace terénu pomocí řídicích bodů plátů a pomocí výškové mapy je však zcela odlišná. Je možné převést terén z plátů na výškovou mapu, opačný převod však možný obecně není.

Převod terénu z plátů na výškovou mapu můžeme uskutečnit takto:

- a) V půdorysu terénu vytvoříme pravidelnou mřížku se stejnými rozměry, jako jsou požadované rozměry výškové mapy.
- b) Každým mřížovým bodem mřížky vedeme přímkou, která je kolmá na půdorys.
- c) Pro každou vedenou přímkou určíme průsečíky s plátem.
- d) Pokud má přímkou právě jeden průsečík s plátem, pak výšku tohoto průsečíku zapiš na výstup.
- e) V případě, že přímkou neprotíná žádný plát, je několik možností jak se zachovat. Jaký postup je nejsprávnější záleží na konkrétním použití.
 - i. Oznamit chybu uživateli a přerušit export
 - ii. Na výstup zapsat hodnotu chybovou NaN (Not a Number)
 - iii. Na výstup zapsat nějakou implicitní hodnotu výšky (třeba 0)

- iv. Na výstup zapsat výšku, kterou má přímce nejbližší bod náležející nějakému plátu
 - v. Na výstup zadat výšku, kterou má nejbližší mřížový bod, který se protíná s nějakým plátem
- f) V případě, že přímka protíná plát vícekrát, existuje také několik možností, jak se zachovat. Nejsprávnější řešení také závisí na konkrétním použití.
- i. Oznámit chybu uživateli a přerušit export
 - ii. Na výstup zapsat maximální z výšek průsečíků
 - iii. Na výstup zapsat minimální z výšek průsečíků
 - iv. Na výstup zapsat průměr z výšek průsečíků
3. Program potřebuje vypočítat detekci kolizí v počítačové hře
- Detekci kolizí se v počítačových hrách používá, když je potřeba například spočítat, zda došlo ke srážce dvou těles nebo co zasáhl náboj vystřelený ze zbraně. Takovéto případy jsou mimo rámec diplomové práce.

Zdroj: (13), (14)

6. Texturování terénu

6.1. Obecné způsoby texturování

Texturu si můžeme představit jako funkci dvou proměnných $T(u, v)$, která k souřadnici v rovině $[u, v]$ přiřadí hodnotu barvy nebo jiné mapované veličiny. Funkce $T(u, v)$ je obvykle dána tabulkou uloženou v paměti počítače. Mapováním textury rozumíme předpis, který pro každý vrchol ve scéně určí souřadnice $[u, v]$ v textuře.

V některých editorech musí uživatel zadávat mapovací funkci ručně pro vrcholy objektu. Takovýto způsob je velmi pracný a náročný na schopnosti uživatele. Jednodušší pro použití nezkušeným uživatelem jsou automatické způsoby generování souřadnic textury.

6.2. Automatické generování souřadnic textury

Texturovací souřadnice $[u, v]$ je možné automaticky vygenerovat z polohových souřadnic vrcholu. Pro generování je možné použít dvou postupů:

V obou případech předpokládáme, že máme k dispozici dvourozměrnou periodickou texturu o velikosti periody u_p a v_p .

- Necht $[x, y, z]$ je souřadnice vrcholu, pro který se mají určit souřadnice textury, kde směr y směřuje vzhůru kolmo na rovný terén. Potom souřadnice textury $[u, v]$ určíme podle rovnice 11.

$$\begin{aligned}u &= x \bmod u_p \\v &= z \bmod v_p\end{aligned}$$

Rovnice 11: Generování souřadnic textury z polohy

- Necht $[t_1, t_2]$ jsou hodnoty parametrů z rozsahu $\langle 0,1 \rangle$ při vykreslování parametrické plochy. Souřadnice textury můžeme určit z parametrů plochy podle rovnice 12.

$$\begin{aligned}u &= t_1 \cdot u_p \\v &= t_2 \cdot v_p\end{aligned}$$

Rovnice 12: Generování souřadnic textury z parametru

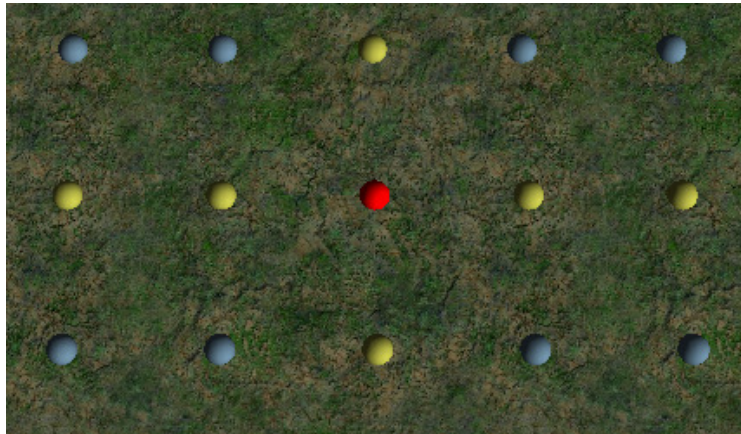
Druhý způsob dává lepší vizuální výsledky. Rozdíly jsou patrné hlavně v případě, že v terénu vznikají převisy. V editoru, který byl vytvořen v rámci diplomové práce, je proto použita druhá metoda.

Zdroj: (15), (16)

6.3. Rozdíl mezi spojitostí C^1 a G^1 při texturování

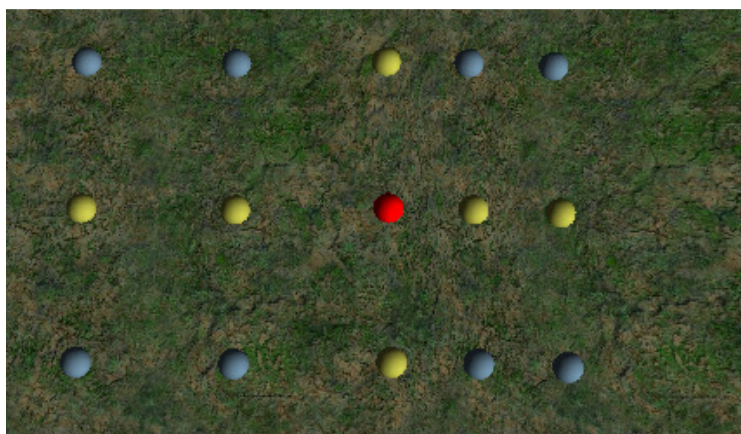
Při texturování hraje důležitou roli stupeň spojitosti, kterým na sebe navazují texturované pláty. Při dosažení parametrické spojitosti C^1 jsou výsledky texturování vizuálně pěkné. Pokud je spojitost pouze geometrická G^1 , jsou na terénu patrné drobné nesrovnalosti. Tvzení budu demonstrovat na snímcích z programu Editor terénů, který vznikl v rámci diplomové práce. Přejít mezi pláty na obrázcích nastává uprostřed snímku.

Na obrázku 8 je vidět, že při C^1 spojitosti není přechod mezi pláty patrný.



Obrázek 8: Spojení čtyř Béziových plátů s C^1 spojitostí

Na obrázku 9 je zobrazen přechod mezi pláty, které mají pouze G^1 a nikoliv C^1 spojitost. Oproti předchozímu obrázku si všimněte změny v texturování v oblasti přechodu. Rychlost texturování se mění skokovitě na přechodu, který je vizuálně znát. Proto je lepší při spojování plátů zajistit C^1 spojitost.



Obrázek 9: Spojení čtyř Béziových plátů s G^1 spojitostí

Zdroj: (15), (16)

7. Výběr grafické knihovny

V současné době existují dvě hlavní knihovny pro trojrozměrnou grafiku: **OpenGL** a **DirectX**. Kromě těchto dvou grafických knihoven existují ještě knihovny, které jsou nadstavbami nad některou z uvedených knihoven. Jedním z úkolů před vývojem editoru pro modelování terénů bylo vybrat vhodnou knihovnu pro vykreslování trojrozměrné grafiky.

7.1. Knihovna OpenGL

Grafická knihovna Open Graphics Library, která se často zkracuje jako OpenGL, představuje standardizované rozhraní pro vykreslování dvojrozměrné i třírozměrné grafiky. Rozhraní Open Graphics Library vytvořila společnost Silicon Graphics v roce 1992. Pro rozhraní OpenGL existují implementace pod operačními systémy Microsoft Windows, Apple Mac OS, Linux i Unix. Program napsaný s použitím OpenGL lze teoreticky spustit na všech těchto platformách. V praxi je to však komplikovanější, jelikož OpenGL obsahuje pouze příkazy pro vykreslování grafiky a třeba příkaz pro vytvoření okna, do kterého se bude kreslit, je už závislý na platformě. Dalším problémem v přenositelnosti může být to, že na některých platformách není implementace rozhraní OpenGL součástí operačního systému. Bývá tudíž třeba distribuovat program spolu s knihovnou OpenGL nebo popsat v manuálu k programu, jak má uživatel knihovnu stáhnout z webu a nainstalovat.

Knihovna OpenGL skládá scénu z jednoduchých tvarů – primitiv. Jejich kombinací pak vznikají složitější tělesa.

Nevýhodou této knihovny OpenGL je to, že způsob programování s ní je z pohledu dnešního programátora poněkud zastaralý. Knihovna byla navržena v době, kdy objektově orientované programování nebylo rozšířené v komunitě počítačových grafiků. Proto také rozhraní OpenGL nebylo navrženo tak, aby využívalo bohatých možností objektově orientovaného programování, které jsou u současných programátorů velmi používané a populární. Poměrně nedostatečná je i typová kontrola vstupních parametrů funkcí. Funkce knihovny OpenGL často vyžadují jako vstupní parametr hodnotu z nějakého výčtu. Protože však se v knihovně nepoužívají výčtové typy, vše je řešeno celočíselnými konstantami. Protože je však v OpenGL celočíselných konstant velké množství, snadno se stane, že se programátor splete a použije konstantu, která je v daném kontextu nepřípustná. Taková chyba pak není detekována během kompilace a při běhu se projevuje podivným chováním programu, tudíž je často obtížné ji najít a opravit.

Zdroj: (17), (18), (19)

7.2. Knihovna DirectX

DirectX je sada programátorských knihoven pro práci s multimediálním obsahem. Převážně je určena pro programování her na platformách Windows pro PC a Xbox.

Nejdůležitější knihovny ze sady DirectX jsou:

- I. **DirectDraw**
Knihovna DirectDraw umožňuje programátorům pracovat s dvourozměrnou rastrovou grafikou. Obsahuje také nástroje pro práci s bitmapovými obrázky.
- II. **Direct3D**
Ústřední knihovnou pro práci s 3D grafikou je knihovna Direct3D. Obsahuje rozsáhlou sadu příkazů pro zobrazování a práci s trojrozměrnou grafikou.
- III. **DirectPlay**
Knihovna DirectPlay v sobě obsahuje podporu pro hraní po lokální síti nebo po Internetu.
- IV. **DirectInput**
Tato knihovna umožňuje používání běžných i herních vstupních zařízení, jako je klávesnice, myš, joystick nebo gamepad.
- V. **DirectSound, DirectMusic**
Tyto knihovny umožňují přehrávání zvukových efektů a hudby.

Zdrojový kód využívající knihovnu DirectX se často obtížně čte. Jedním z důvodů je používání prefixů identifikátorů známé též jako „maďarská notace“. (viz Obrázek 10)

```
[Kód pro vytvoření primárního povrchu]

LPDIRECTDRAWSURFACE lpDDSPRimary;
DDURFACEDESC ddsd;
HRESULT ddrval;

//Clear all fields in structure
memset( &ddsd, 0, sizeof(ddsd) );

//Set flags in structure
ddsd.dwSize = sizeof( ddsd );
ddsd.dwFlags = DDSD_CAPS | DDSD_BACKBUFFERCOUNT;
ddsd.ddsCaps.dwCaps = DDSCAPS_PRIMARYSURFACE | DDSCAPS_FLIP |
DDSCAPS_COMPLEX;
ddsd.dwBackBufferCount = 1;

//Create Surface
ddrval = lpDD->CreateSurface( &ddsd, &lpDDSPRimary, NULL );
```

Obrázek 10: Ukázka kódu používajícího DirectX

Zdroj: (20), (21), (22)

7.3. Knihovna Java3D

Knihovna Java3D je nadstavbou nad knihovnami OpenGL a DirectX. Nejedná se pouze rozhraní, které by pouze umožňovalo volat funkce těchto grafických knihoven. Více se podobá moderním grafickým knihovnám pro profesionální tvorbu 3D her.

Knihovna Java3D je založena na vykreslování grafu scény. Graf scény lze se značnou mírou zobecnění považovat za skript, podle kterého se scéna vykresluje. Graf scény se dá během běhu programu libovolně měnit a tím ovlivňovat scénu způsobem, který by pomocí tradičních metod nebyl možný.

Při změnách scény za běhu programu je vždy nutné najít kompromis mezi rychlostí a dynamičností změn. Například pokud ve scéně transformujeme objekt pomocí několika transformačních matic, můžeme transformační matice sloučit (součinem matic) do jedné a tím urychlit běh programu. Tuto operaci však je možné však udělat pouze tehdy, pokud víme, že transformační matice nebudou při běhu programu měněny. Pokud bychom chtěli maximálně dynamický systém, který by uměl za běhu měnit libovolnou transformační matici, pak bychom museli vypustit veškeré optimalizace transformačních matic a tím zpomalit vykreslování. (Transformační matice jsou zde uvedeny jenom jako příklad. Ve scéně je mnohem více případů, kdy lze optimalizovat. Pokud například máme nepohyblivé zdroje osvětlení a nepohyblivé objekty, můžeme osvětlení scény vypočítat pouze jednou na začátku místo pomalejšího počítání osvětlení v každém snímku.)

V knihovně Java3D je vykreslování vyřešeno zároveň velmi efektivně i dynamicky. Při vytváření grafu scény uživatel u objektů ve scéně zadá, jaké změny jsou povoleny pomocí systému způsobilostí (capabilities).

Zdroj: (23), (24)

7.3.1. Kritika knihovny Java3D

Kromě výhod knihovny Java3D lze nalézt i některé kritické ohlasy. Například v knize (25) se jich objevuje několik. Výhrady zde uvedu s vysvětlení, proč daná věc není při realizaci práce na obtíž. Nakonec uvádím svoje zkušenosti, zda jsem se s danou nevýhodou opravdu setkal.

1. Grafické programy v Javě nevyužívají grafického akcelérátoru, a proto jsou pomalé

Tento argument není pravdivý. Mnoho mých kolegů se domnívalo, že když je knihovna Java3D platformově nezávislá, nemůže využívat grafický akcelérátor, protože způsob, jakým se využívá grafický akcelérátor, je závislý na platformě. Bylo by sice možné do knihovny zahrnout kód pro ovládání grafického akcelérátoru na běžně rozšířených platformách, ale potom by byla aplikace velmi rozsáhlá a to by značně komplikovalo distribuci aplikace. Jednalo by se o řádově stovky MB, což není příliš při distribuci na optických discích, ale při stahování z Internetu je to již velká překážka.

Knihovna Java3D však takovými problémy netrpí. Není totiž zcela platformově nezávislá. Pro každou běžně rozšířenou platformu existuje samostatná verze knihovny, která obsahuje pouze kód pro ovládání grafického akcelerátoru na příslušné knihovně. Při distribuci pomocí optického disku lze přiložit knihovny pro všechny platformy. Při distribuci pomocí Internetu stačí uživateli, aby si stáhl pouze knihovnu určenou pro jeho platformu.

2. Programy napsané s použitím knihovny Java3D mají nižší výkon.

Podle srovnání výkonnosti programů napsaných pomocí C++ a OpenGL oproti programům napsaných pomocí Javy a Javy3D vyplývá, že programy napsané v C++ jsou asi 2 krát rychlejší. Mohlo by se zdát, že C++ je pro programování grafických aplikací výhodnější, musíme však srovnávat i podle jiných parametrů. Program napsaný v Javě/Javě3D má přibližně třikrát méně řádků kódu než program se stejnou funkcionalitou napsaný v C++/OpenGL. V diplomové práci jsem vytvořil editor terénů, ve kterém nebyl požadavek na maximální rychlost. Současné počítače jsou natolik rychlé, že bez problémů dorovnají zpomalení Javy, programy napsané v Javě tak běží stejně rychle jako programy napsané v C++ na počítačích, které byly běžné zhruba před 18 měsíci. Naopak menší velikost kódu pro dosažení stejné funkcionality je velmi vítanou vlastností.

Moje zkušenosti: Během programování diplomové práce jsem měl jenom několik málo problémů s rychlostí aplikace. Většinou se však nejednalo o problémy způsobené knihovnou Java3D. Jednalo se o případy, kdy zpracovávám celý terén – při exportu terénu, při transformacích celého terénu nebo při automatických korekcí terénu. Takovéto operace se však s terénem provádějí jen zřídka. S problémy jsem se většinou vypořádal implementací efektivnějšího algoritmu a omezením alokací nových objektů na haldě.

3. Knihovna Java3D umožňuje pouze práci na příliš vysoké úrovni.

Moje zkušenosti: Problém s vysokou úrovní Javy jsem měl pouze jednou. Pro stínování terénu jsem chtěl původně použít Phongův stínovací model, který počítá osvětlení v každém vykreslovaném pixelu. Tato metoda je poměrně náročná na grafický hardware a nedá se v reálném čase emulovat softwarem. Nakonec jsem se rozhodl, že použiji jednodušší Gourandův stínovací model, který počítá intenzitu osvětlení jenom ve vrcholech polygonů a intenzitu ve vnitřních bodech polygonu získá interpolací hodnot ve vrcholech.

4. Hry vytvořené pomocí knihovny Java3D nelze spustit na herních konzolách.

Je pravda, že zatím žádná herní konzole nepodporuje Javu. Úkolem diplomové práce však je vytvořit editor pro modelování terénu na osobním počítači a není požadováno, aby šla aplikace spustit na herních konzolách. Pokud v editoru bude vymodelován terén, je možné ho vyexportovat a následně použít jako data pro počítačovou hru. Vyexportovaný terén je možné použít i na herních konzolách.

Moje zkušenosti: Jako cílovou platformu pro svoji aplikaci jsem vybral Windows XP 32-bit a Windows Vista 32-bit. Přesto jsem programoval vždy tak, aby byl kód nezávislý na platformě a používal pouze platformově nezávislé knihovny. Výsledný program není počítačovou hrou, a tudíž není třeba ho spouštět na herních konzolách.

Zdroj: (25), (26)

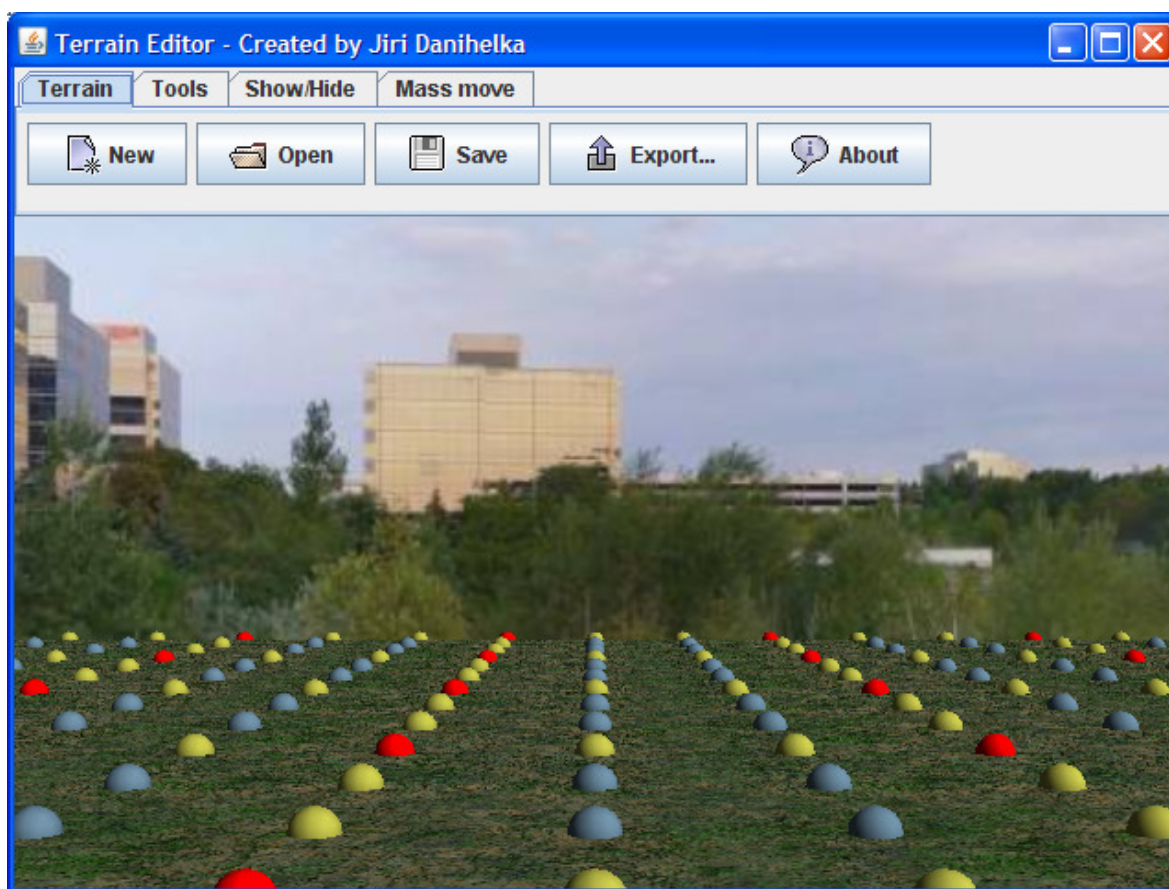
8. Vytvořený program Editor terénů

8.1. Obecný popis programu

Jako součást diplomové práce vznikl program nazvaný Editor terénů (Obrázek 11). Tento software je určen pro demonstraci metod navržených v diplomové práci. Jedná se o samostatný editor pro trojrozměrné modelování, program má však hlavně demonstrovat navržené metody a na ostatní funkčnost nebyl kladen tak velký důraz.

Hlavním cílem programu je ukázat, že terén je možné snadno a přirozeně modifikovat pomocí metody přesouvání řídicích bodů u Bézierových plátů.

Program vznikl na základě požadavků pana ing. Varnušky PhD., který je externím konzultantem diplomové práce. Pan Varnuška se zabývá programováním nástrojů pro tvorbu trojrozměrného virtuálního prostředí pro počítačové hry ve firmě Black Element Software. Zajímalo ho, zda by bylo možné vytvořit nástroj, pomocí kterého by se dal snadno modelovat terén pro počítačové hry. Odpověď na jeho otázky se snaží dát program Editor terénů, který v sobě obsahuje nástroje pro snadné a intuitivní modelování terénu. Pokud se metoda osvědčí, může být implementována jako zásuvný modul do stávajících profesionálních nástrojů na tvorbu her.



Obrázek 11: Program Editor terénů

8.2. Instalace a spuštění Editoru terénů

8.2.1. Platformy Windows XP 32 bit, Windows Vista 32 bit

Pro spuštění programu je potřeba mít nainstalovanou Javu (Java Runtime Environment - JRE) alespoň ve verzi 1.5. Jinak není potřeba nic instalovat.

Java Runtime Environment lze zdarma stáhnout na adrese:

<http://java.sun.com/javase/downloads/index.jsp>

Program spustíte dvojklikem na souboru `TerrainEditor.jar`.

Pokud nemáte příponu `*.jar` asociovanou s Javou, napište do konzole:

```
java -jar TerrainEditor.jar
```

Pokud by se program nespustil, zkontrolujte, zda se přiložené `dll` knihovny nachází ve stejném adresáři jako archiv `jar`.

8.2.2. Ostatní platformy Windows

Pokuste se program spustit podle postupu pro Windows XP a Vista. Pokud se vám to nepodaří, postupujte podle návodu pro ostatní platformy.

8.2.3. Ostatní platformy – Linux, Solaris, Mac Os

Pro spuštění programu je potřeba mít nainstalovanou Javu (Java Runtime Environment - JRE) alespoň ve verzi 1.5. Dále je potřeba mít nainstalovanou knihovnu Java3D.

Java Runtime Environment lze zdarma stáhnout na adrese:

<http://java.sun.com/javase/downloads/index.jsp>

Knihovnu Java3D lze zdarma stáhnout na adrese:

<http://java.sun.com/products/java-media/3D/download.html>

Program spustíte pomocí následujícího příkazu:

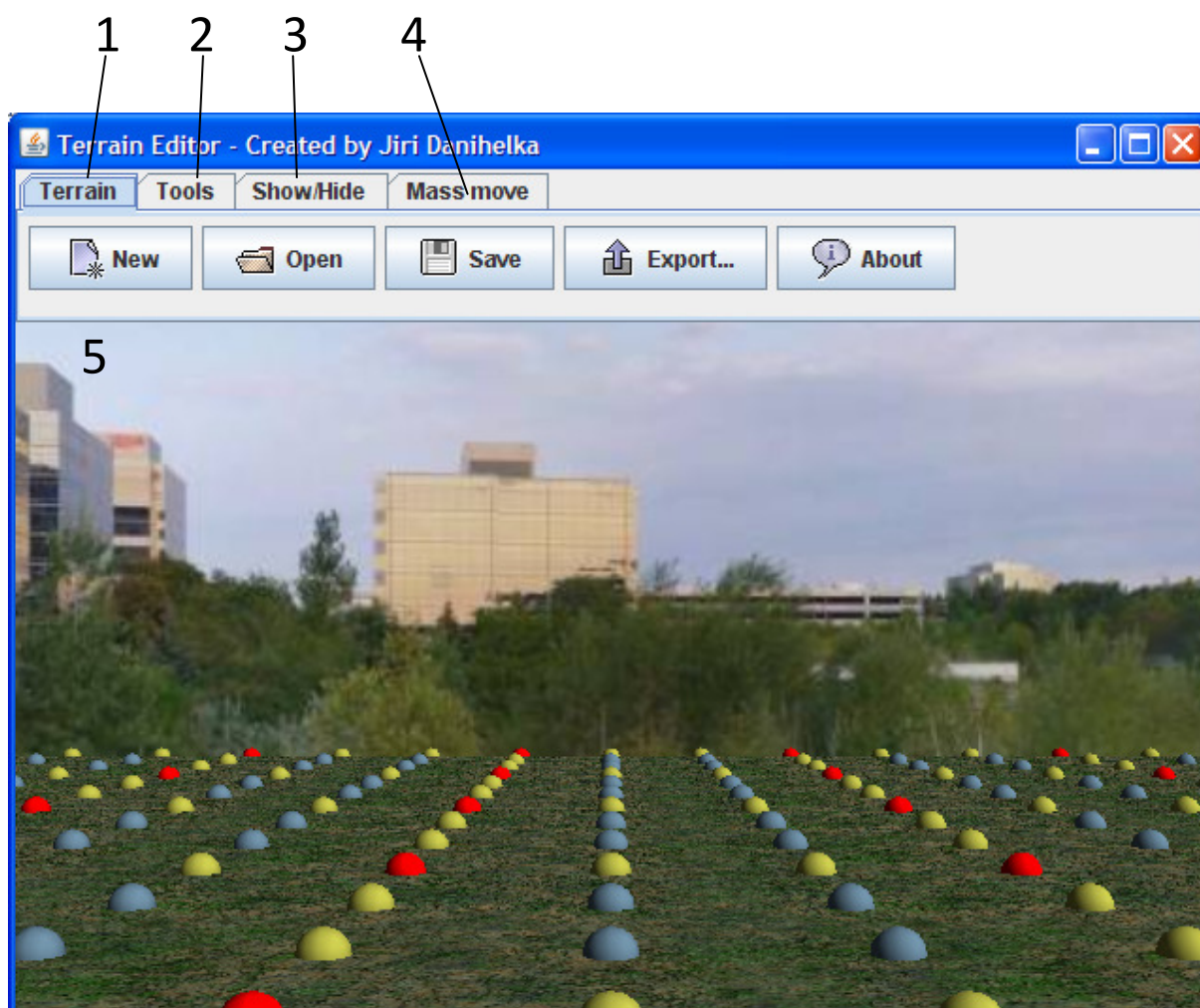
```
java -jar TerrainEditor.jar
```

9. Ovládání Editoru terénů

Při návrhu Editoru terénů byl kladen velký důraz na snadné a intuitivní ovládání. Díky tomu zvládne pracovat s Editorem terénů i málo zkušený uživatel. V následujících podkapitolách je uveden stručný popis ovládání Editoru terénů.

9.1. Hlavní okno Editoru terénů

Po spuštění Editoru terénů se na obrazovce objeví hlavní okno programu (Obrázek 12). Hlavní okno se skládá z několika částí, které jsou popsány dále.



Obrázek 12: Hlavní okno Editoru terénů

1. Záložka Terén (Terrain)
Záložka Terén obsahuje hlavní ovládací prvky pro práci s editorem a se soubory.
2. Záložka Nástroje (Tools)
Tato záložka obsahuje nástroje, pomocí kterých může uživatel terén editovat.

3. Záložka Ukázat/Schovat (Show/Hide)
Záložka obsahuje nastavení, které ovládací prvky se mají zobrazit v editačním okně.
4. Záložka Hromadný přesun (Mass move)
Tato záložka slouží k nastavení nástroje pro hromadný přesun řídicích bodů
5. Editační okno
Největší část hlavního okna zabírá editační okno, které zobrazuje editovaný terén.
Nejvíce operací s terénem uživatel provádí v editačním okně.

9.2. Ovládání kamery v Editoru terénů

Kameru je možné ovládat buď pomocí myši, nebo pomocí klávesnice.

9.2.1. Ovládání kamery pomocí myši

Ovládání kamery pomocí myši je hlavní způsob ovládání kamery.

- a) Pokud uživatel stiskne pravé tlačítko kdekoliv v editačním okně a se stisknutým pravým tlačítkem pohybuje myší, podle pohybu myši se v reálném čase bude měnit směr pohledu kamery.
- b) Pokud uživatel otočí kolečkem myši nahoru, kamera se posune směrem dopředu.
- c) Pokud uživatel otočí kolečkem myši dolů, kamera se posune směrem dozadu.
- d) Pokud uživatel umístí ukazatel myši na některý z řídicích bodů a stiskne prostřední tlačítko myši (kolečko myši), kamera změní svůj směr pohledu tak, aby zvolený řídicí bod byl ve středu editačního okna.
- e) Pokud uživatel umístí ukazatel myši na některý z řídicích bodů a stiskne prostřední tlačítko myši (kolečko myši), kamera změní svůj směr pohledu tak, aby zvolený řídicí bod byl ve středu editačního okna. Pokud bude uživatel dále pohybovat myší se stisknutým prostředním tlačítkem, kamera bude obíhat kolem zvoleného řídicího bodu podle pohybů myši. Zvolený řídicí bod přitom stále bude ve středu editačního okna.
- f) Pokud uživatel umístí ukazatel myši na některý z řídicích bodů a kolečkem myši otočí nahoru, kamera se přiblíží k zvolenému řídicímu bodu.
- g) Pokud uživatel umístí ukazatel myši na některý z řídicích bodů a kolečkem myši otočí dolů, kamera se oddálí od zvoleného řídicího bodu.

9.2.2. Ovládání kamery pomocí klávesnice

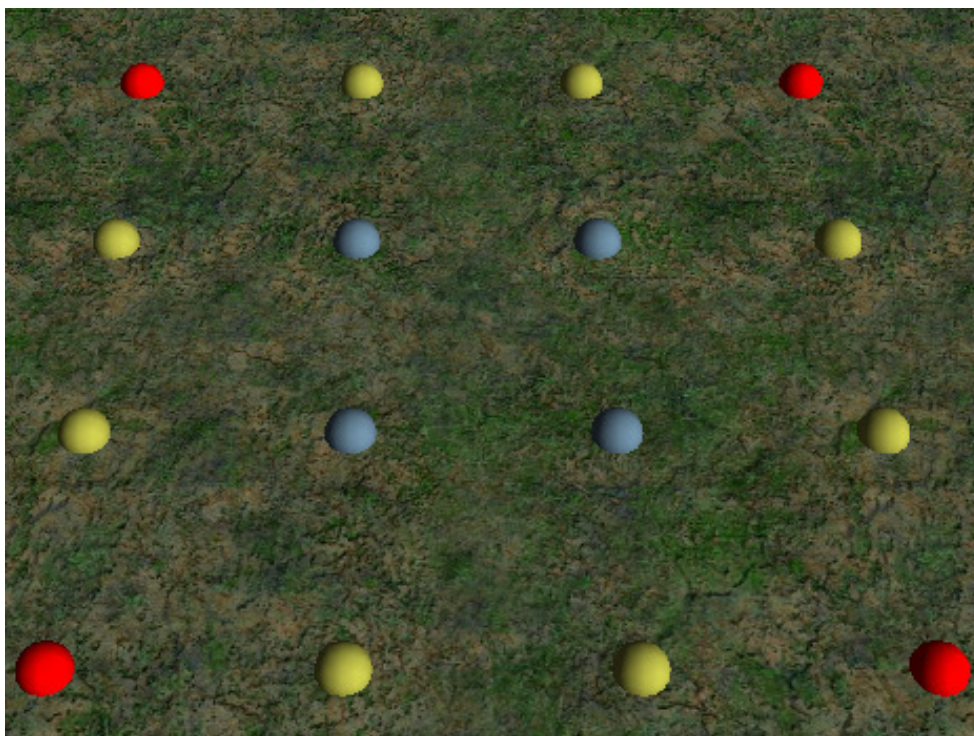
Pomocí kurzorových kláves je možné posouvat kamerou. Ovládání kamery pomocí klávesnice je pouze doplňkovým způsobem. Pro ovládání Editoru terénů je potřeba myš a klávesnicí ji nelze nahradit.

9.3. Manipulace s Bézierovými pláty

V Editoru terénů může uživatel manipulovat s Bézierovými pláty pomocí šestnácti řídicích bodů, které jsou na začátku uspořádány do mřížky 4×4 (Obrázek 13).

Podle barvy je rozlišen druh řídicího bodu:

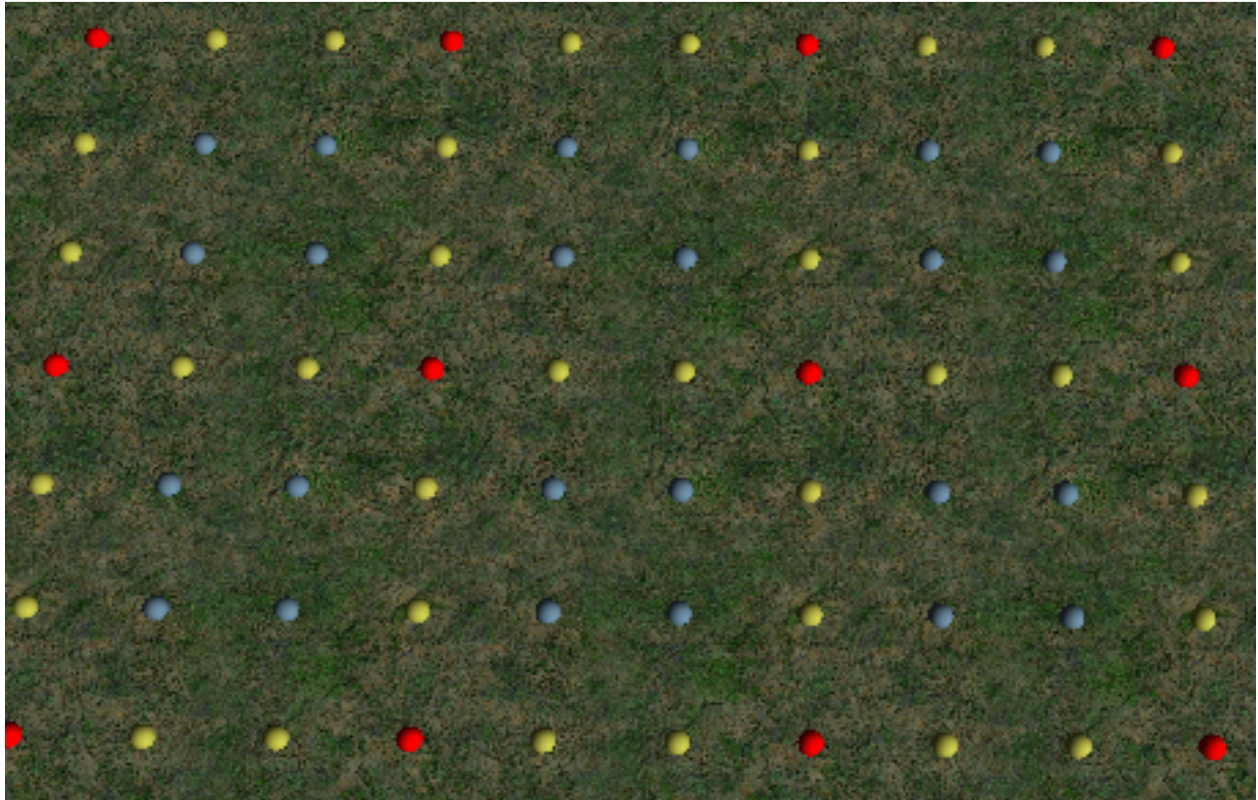
- Červenou barvou jsou zobrazeny řídicí body, které jsou v rozích Bézierova bikubického plátu. Tyto body jsou interpolační a terén jimi vždy prochází.
- Žlutou barvou jsou zobrazeny řídicí body na okrajích Bézierova plátu. Žluté řídicí body jsou aproximační, to znamená, že terén jimi obecně neprochází. V počátečním postavení bodů po zapnutí Editoru terénů, je však plát v takovém stavu, že prochází všemi řídicími body, tedy i aproximačními.
- Modré řídicí body jsou ve prostřed Bézierova bikubického plátu. Modré řídicí body jsou také aproximační a terén jimi v obecném případě neprochází.



Obrázek 13: Bézierův plát v Editoru terénů

9.3.1. Sdílení řídicích bodů mezi Bézierovými pláty

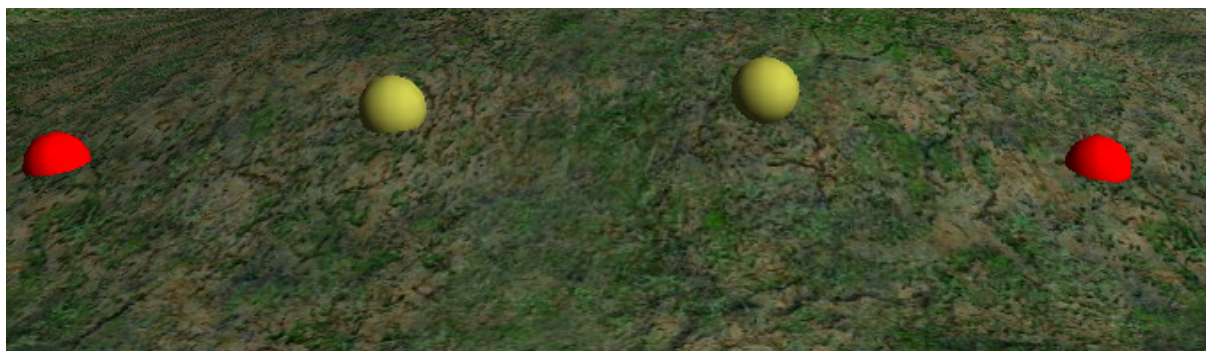
Na obrázku 14 je vidět šest bikubických Bézierových plátů. Všimněte si, že některé řídicí body jsou sdílené mezi pláty. Červené řídicí interpolační body v rozích Bézierových plátů jsou sdílené čtveřicí sousedících Bézierových plátů. Žluté řídicí body na hranách plátů jsou sdílené dvojicí Bézierových plátů. Modré řídicí body uvnitř Bézierových plátů sdílené nejsou.



Obrázek 14: Sousedící Bézierovy pláty v Editoru terénů

Sdílení řídicích bodů mezi pláty zaručuje C^0 spojitost. Podmína pro C^0 spojitost je zároveň nutnou podmínkou pro spojitosti vyšších ráďů. Editor terénů tudíž vždy zajišťuje, že bude zachován spojitost C^0 .

Bikubické Bézierovy pláty jsou spojeny pomocí kubických Bézierových křivek. Na obrázku 15, který zobrazuje řídicí body pro hraniční křivku mezi Bézierovými pláty, si všimněte, že terén prochází červenými interpolačními řídicími body a neprochází žlutými aproximačními řídicími body.



Obrázek 15: Řídící body pro hraniční Bézierovu křivku

Pokud bychom pomocí této čtveřice řídicích bodů zkonstruovali kubickou Bézierovu křivku, celá křivka by ležela přesně mezi dvojicí Bézierových bikubických plátů, které sdílejí zobrazené žluté řídicí body. Tato křivka by oba Bézierovy pláty od sebe oddělovala.

9.3.2. Manipulace s řídicími body Bézierových plátů

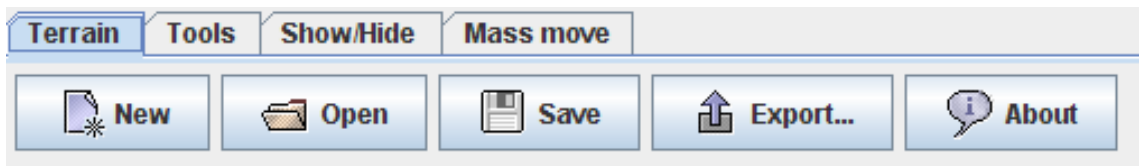
Uživatel může s řídicími body Bézierových plátů manipulovat pomocí myši.

- a) Pokud uživatel umístí ukazatel myši na některý z řídicích bodů a stiskne levé tlačítko myši, vybraný řídicí bod tak uchopí. Řídící bod zůstane uchopený tak dlouho, dokud bude levé tlačítko myši stisknuto.
- b) Pokud uživatel uchopí bod a bude pohybovat myší, uchopený řídicí bod se bude pohybovat společně s kurzorem myši. Vzdálenost uchopeného řídicího bodu od kamery přitom bude zůstávat stejná.
- c) Pokud uživatel uchopí bod a bude otáčet kolečkem myši nahoru, uchopený řídicí bod se bude vzdalovat od kamery.
- d) Pokud uživatel uchopí bod a bude otáčet kolečkem myši dolů, uchopený řídicí bod se bude přibližovat ke kameře.

Při manipulaci s řídicími body bude zachována C^1 spojitost, pokud uživatel zachovávání C^1 spojitosti předtím nevynechá.

9.3.3. Popis ovládacích prvků záložky Terrain

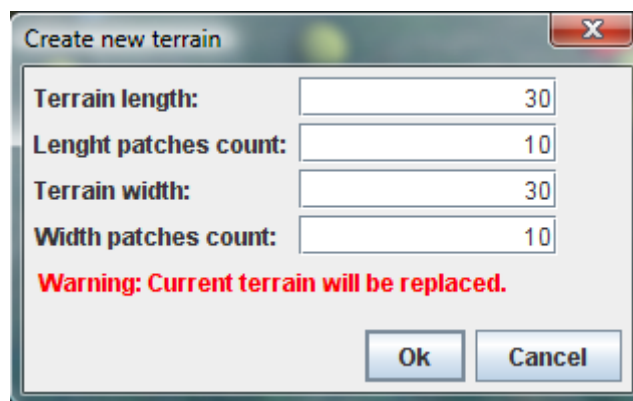
Záložka **Terrain** (Obrázek 16) obsahuje ovládací prvky pro práci s terénem jako celkem.



Obrázek 16: Záložka Terrain

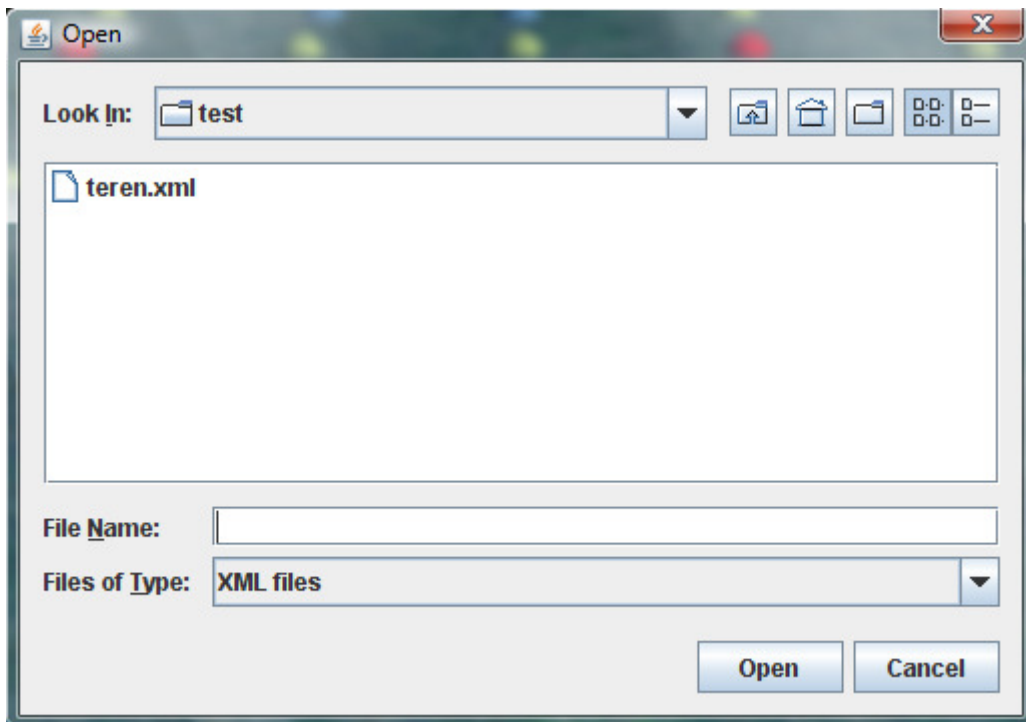
Na záložce se nachází několik tlačítek, jejich význam je následující:

- a) **Tlačítko New** zobrazí dialogové okno pro vytvoření nového terénu (viz Obrázek 17). V dialogovém okně může uživatel nastavit, délku a šířku nového terénu a také na kolik Bézierových plátů má být terén rozdělen v jednotlivých směrech. Po potvrzení dialogu je vytvoří nový terén, který bude zcela rovný a předcházející terén bude nahrazen novým terénem. Tuto funkci by měl uživatel použít, jen pokud stávající terén již nebude potřebovat.

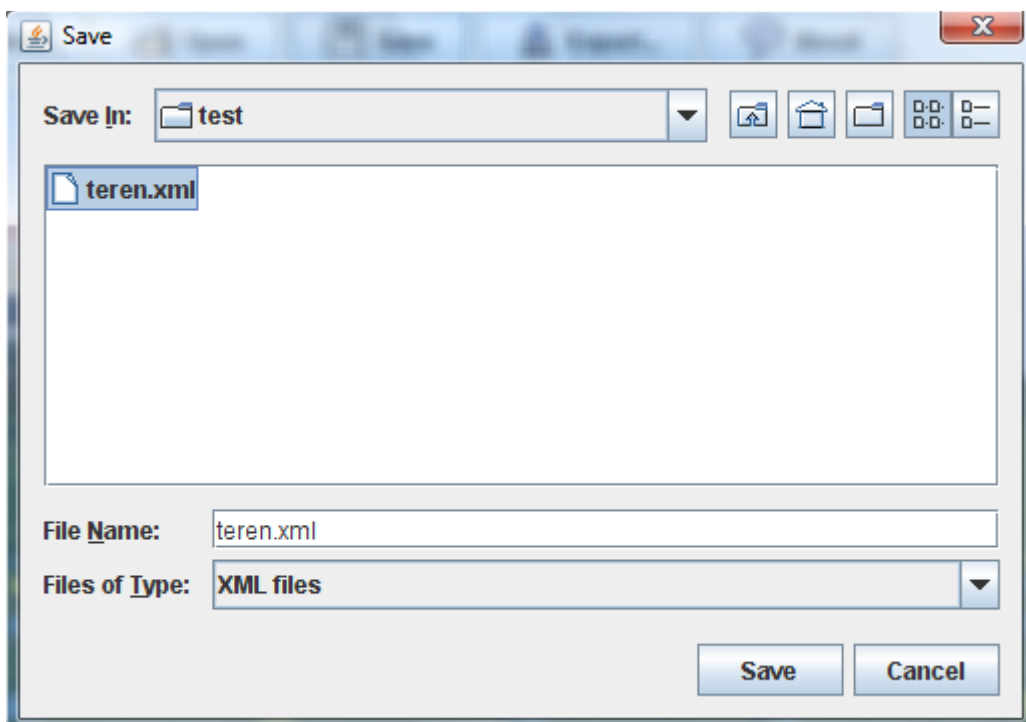


Obrázek 17: Dialogové okno pro vytvoření nového terénu

- b) **Tlačítko Open** slouží k načtení terénu ze souboru. Terén se načítá z formátu XML. Po stisknutí se zobrazí dialogové okno pro otevření souboru (Obrázek 18).
- c) **Tlačítko Save** slouží k uložení terénu do souboru. Terén se ukládá do formátu XML. Po stisknutí se zobrazí dialogové okno pro uložení souboru (Obrázek 19).
- d) **Tlačítko Export** slouží k převodu terénu do výškové mapy ve formátu RAW. Formát RAW lze následně otevřít v programu Photoshop.
- e) **Tlačítko About** zobrazí dialogové okno s informacemi o programu a autorovi.



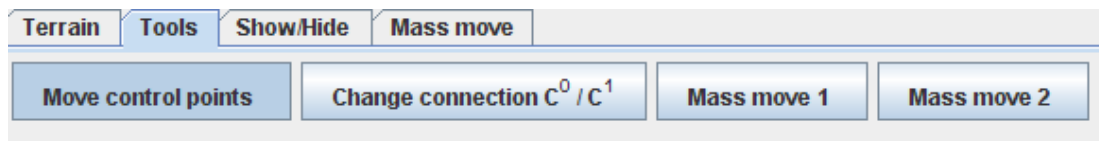
Obrázek 18: Dialogové okno pro otevření souboru s terémem



Obrázek 19: Dialogové okno pro uložení souboru s terémem

9.3.4. Popis ovládacích prvků záložky Tools

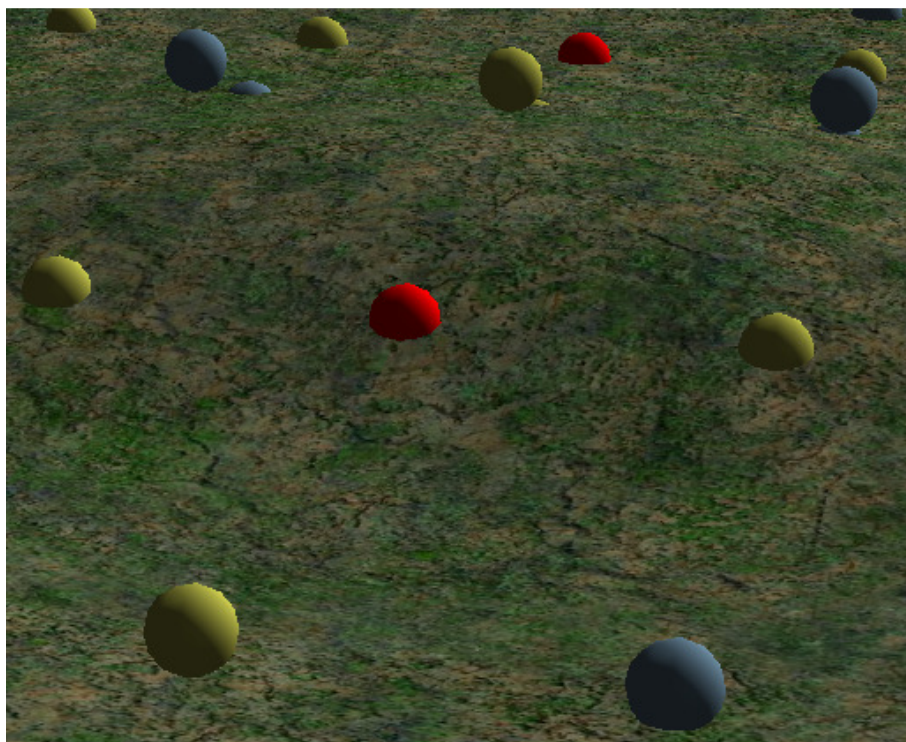
Na záložce Tools (Obrázek 20) se nacházejí nástroje pro editaci terénu.



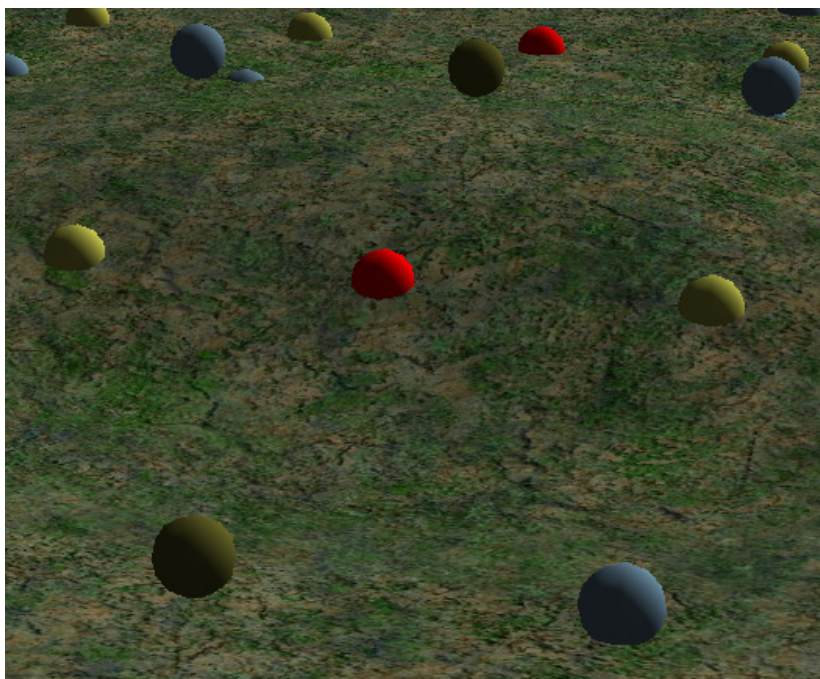
Obrázek 20: Záložka Tools

Nástroj **Move control points** – Základním nástrojem pro editování terénu je přesouvání řídicích bodů. Pomocí tohoto nástroje může uživatel pomocí myši uchopit některý z řídicích bodů a pomocí pohybu myši ho přesunout. Posouvání řídicích bodů pomocí myši již bylo popsáno dříve.

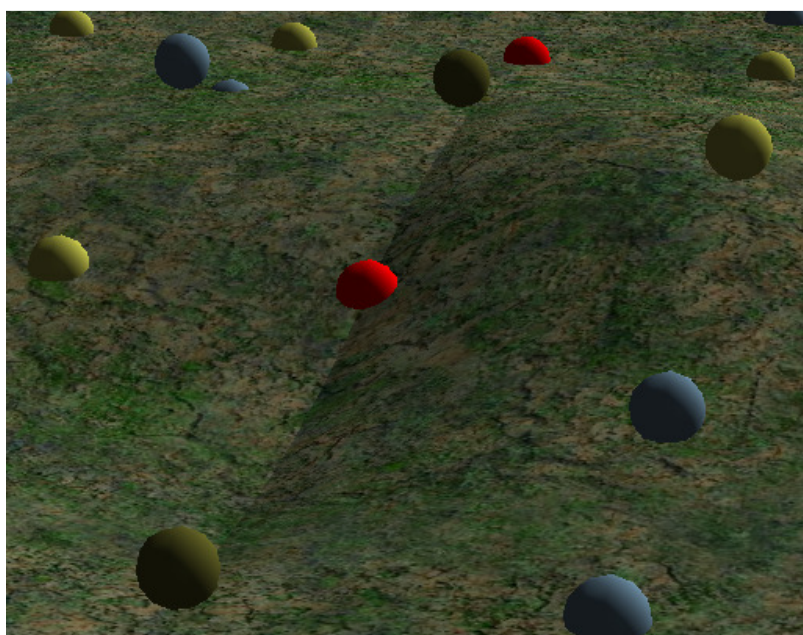
Nástroj **Change connection** slouží k označení míst, kde uživatel chce, aby byl nižší stupeň spojitosti. Editor terénu zajišťuje, že mezi pláty bude C^1 spojitost (viz Obrázek 21). Pokud uživatel chce mezi některými pláty nižší stupeň spojitosti, může pomocí nástroje Change connection označit rozhraní mezi pláty, na kterém chce vytvořit ostrý zlom. Rozhraní plátů označí stisknutím tlačítka myši na žlutém řídicím bodě, který reprezentuje hranici mezi pláty. Žluté řídicí body pak změny baru na tmavší jako znamení nižší spojitosti (viz Obrázek 22). Pokud uživatel následně bude přesouvat řídicí body v okolí oblasti s nižší spojitostí, vytvoří se ostrý zlom mezi pláty (viz Obrázek 23).



Obrázek 21: Původní stav - terén s C^1 spojitostí



Obrázek 22: Terén se sníženou spojitostí



Obrázek 23: Ostrý zlom vzniklý kvůli snížení třídy spojitosti

Pomocí nástroje Change connection je možné změnit spojitost i na větší oblasti najednou. Pokud uživatel stiskne levé tlačítko myši na některém z řídicích bodů a se stisknutým levým tlačítkem nastaví ukazatel myši na jiný řídicí bod a levé tlačítko uvolní, pak se změní spojitost v obdélníkové oblasti, kterou uživatel vybral pohybem myši se stisknutým levým tlačítkem.

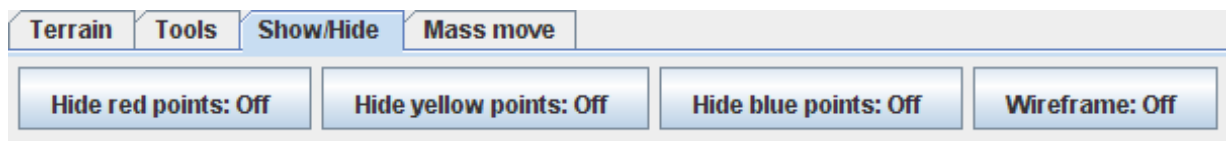
Na záložce Tools se nachází dva nástroje pro hromadné přesouvání bodů - **Mass move 1** a **Mass move 2**.

Nástroj Mass move 1 přesouvá body, které se nacházejí v kruhové oblasti. S nástrojem Mass move 1 uživatel vybere řídicí bod, který bude středem kruhové oblasti a stiskne na něm levé tlačítko. Pohybem myši se stisknutým levým tlačítkem může uživatel pohybovat vybraným řídicím bodem a ostatními řídicími body v kruhovém okolí vybraného řídicího bodu.

Nástroj Mass move 2 také umožňuje hromadně přesouvat řídicí body. Pokud uživatel stiskne levé tlačítko myši na některém z řídicích bodů a se stisknutým levým tlačítkem nastaví ukazatel myši na jiný řídicí bod a levé tlačítko uvolní, pak se označí body v obdélníkové oblasti, kterou uživatel vybral pohybem myši se stisknutým levým tlačítkem. Pokud uživatel následně stiskne levé tlačítko myši na některém z vybraných řídicích bodů, pohybem myši se stisknutým levým tlačítkem může uživatel pohybovat vybranými řídicími body.

9.3.5. Popis ovládacích prvků záložky Show/Hide

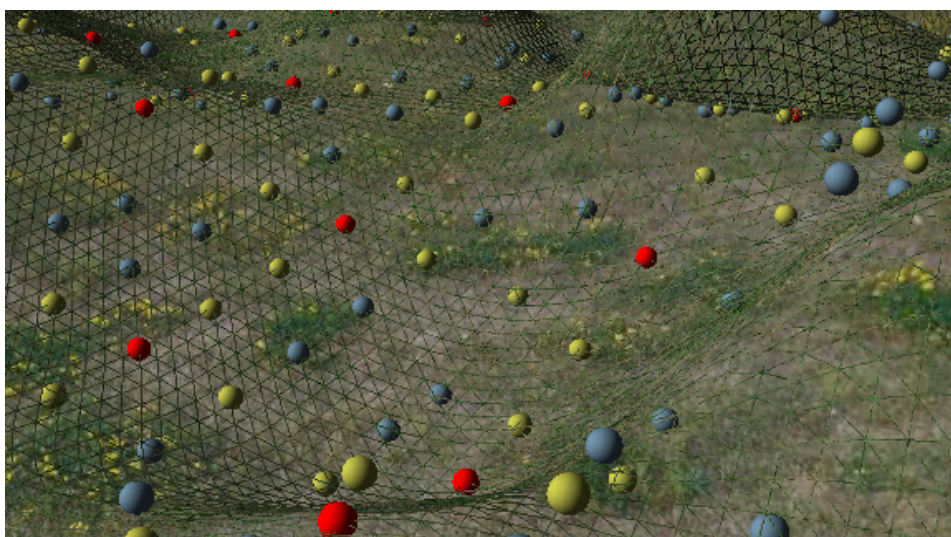
Záložka Show/Hide je zobrazena na obrázku 24.



Obrázek 24: Záložka Actions

Pomocí tlačítek **Hide red points**, **Hide yellow points** a **Hide blue points** může uživatel vypnout nebo zapnout vykreslování červených, žlutých nebo modrých řídicích bodů. Uživatel si vypnutím některých řídicích bodů může zjednodušit modelování terénu.

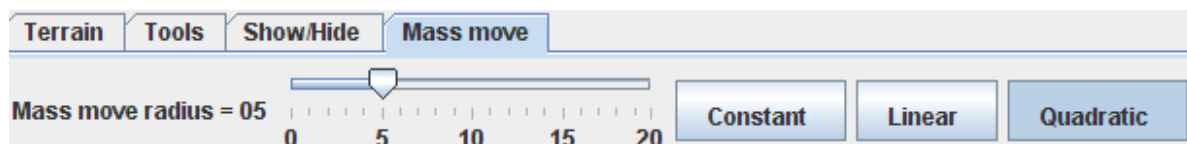
Pomocí tlačítka **Wireframe** může uživatel zapnout vykreslování terénu ve formě drátového modelu. (viz Obrázek 25) Zobrazení drátového modelu může být výhodné například když některé řídicí body jsou pod povrchem terénu a uživatel na ně nevidí. Drátový model je průhledný a tudíž jsou viditelné i řídicí body pod povrchem.



Obrázek 25: Drátěný model terénu

9.3.6. Popis ovládacích prvků záložky Mass move

Záložka Mass move je zobrazena na obrázku 26.



Obrázek 26: Záložka Mass move

Pomocí záložky Mass move může uživatel nastavit parametry pro nástroj Mass move 1.

Pomocí posuvníku se dá nastavit poloměr oblasti, se kterou bude nástroj pro hromadný posun bodů pracovat. Tím je možné rozšířit nebo zúžit oblast, která je nástrojem pro hromadný posun bodů ovlivněna.

Na záložce se dále nachází tlačítka Constant, Linear a Quadratic, kterými se dá nastavit režim nástroje hromadného přesouvání bodů.

V režimu Quadratic (Obrázek 27) klesá vliv na posunutí bodu s druhou mocninou vzdálenosti od středu oblasti. Výsledkem používání nástroje v tomto režimu jsou kopce ve tvaru rotačního paraboloidu.

V režimu Linear (Obrázek 28) klesá vliv na posunutí bodu lineárně se vzdáleností od středu oblasti. Výsledkem používání nástroje v tomto režimu jsou ostré hory ve tvaru kuželů.

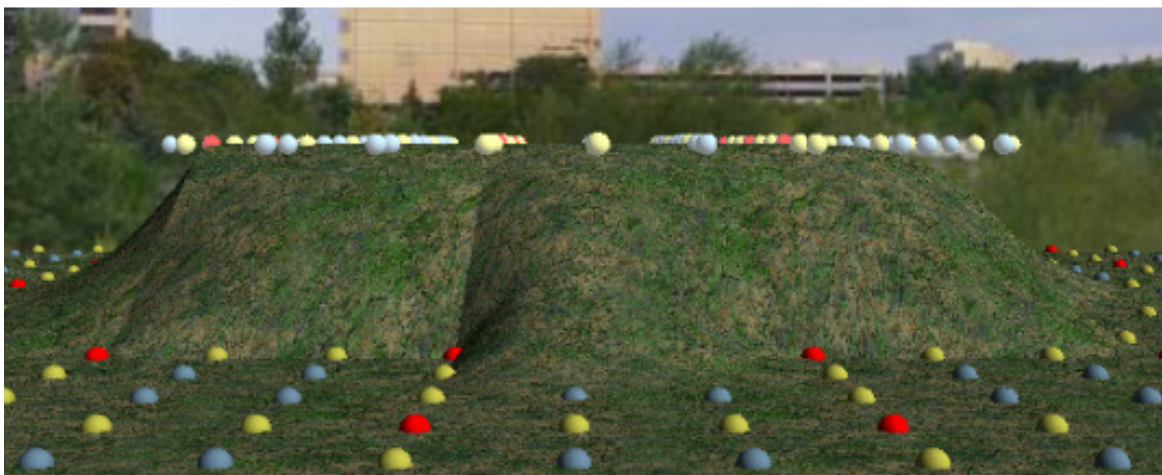
V režimu Constant (Obrázek 29) vliv na posunutí bodu neklesá se vzdáleností od středu oblasti. Výsledkem používání nástroje v tomto režimu jsou vyvýšená místa ve tvaru stolové hory.



Obrázek 27: Režim Quadratic



Obrázek 28: Režim Linear



Obrázek 29: Režim Constant

10. Závěr

V diplomové práci byly porovnány jednotlivé metody modelování terénů. Jako nejlepší byla vybrána metoda založená na přesouvání řídicích bodů u Bézierových plátů.

Vytvořený program Editor terénů v praxi potvrzuje, že navržená metoda je pro vytváření terénů vhodná a pomocí ní se dá snadno terén modelovat. Na základě metody navržené v diplomové práci mohou v budoucnu vzniknout nástroje, které obyčejným lidem i profesionálním modelářům umožní snadné vytváření trojrozměrného obsahu a tak podpořit rychlejší rozšiřování trojrozměrné grafiky ve světě počítačů.

10.1. Vyhodnocení splnění zadání programu

Tato podkapitola srovná dosažené cíle s původně stanovenými požadavky.

Požadavky na produkt

- Práce s editorem bude interaktivní a bude probíhat v reálném čase.
- Při vytváření nového terénu uživatel zvolí rozměry terénu v km a počet plátů, na které bude terén rozdělen.
- Editor bude umět exportovat data v podobě výškové mapy v rozlišení $32 \cdot x + 1$ a normálové mapy v rozlišení $32 \cdot x \cdot \text{SIZE} + 1$, kde x je libovolné přirozené číslo zadané uživatelem. Editor také bude umět exportovat souřadnice pro texturování.
- Editace plátů bude probíhat změnou polohy řídicích bodů plátů v prostoru. Při návrhu GUI, bude kladen důraz na jednoduchost editování terénu.
- Uživatel bude moci nastavit stupeň spojitosti mezi pláty na hodnoty G^0 nebo G^1

Splnění požadavků na produkt

- Podařilo se, aby modelování v editoru bylo interaktivní a v reálném čase. Pan Varnuška si v průběhu vývoje sice stěžoval na pomalejší odezvu programu, protože měl pomalejší počítač, než na kterém byl software vyvíjen. Optimalizacemi kódu se však podařilo tyto obtíže zmírnit.
- Uživatel může vytvářet nový terén, volit jeho rozměry a počet plátů, na které se rozdělí.
- Požadavky pro export dat se změnily v průběhu vývoje. Jako perspektivnější byla vybrána možnost exportovat do formátu RAW.
- Byla nalezena metoda, jak intuitivně přesouvat řídicí body v prostoru a tím modelovat tvar plátů.

- Oproti původnímu zadání uživatel volí spojitost mezi C^0 a C^1 a nikoliv G^0 a G^1 . V průběhu prací se zjistilo, že zajištění parametrické spojitosti je důležité pro texturování. (Geometrické a parametrické spojitosti se věnuje kapitola 3.)

Požadavky na proces

- Vhodný programovací jazyk a grafické knihovny pro práci si zvolí řešitel. Zadavatel v tomto ohledu nemá požadavky. Jazyky .NET nejsou preferované.
- Řešitel se podrobně seznámí s dostupnými křivkami a plochami a vybere vhodné řešení pro použití v editoru.

Splnění požadavků na proces

- Jako programovací jazyk byla zvolena Java s použitím grafické knihovny Java3D. Tím bylo dosaženo větší platformové nezávislosti než byla původně požadována.
- Byla porovnána vhodnost různých druhů ploch pro modelování. Jako nejvhodnější byly vybrány Bézierovy pláty.

Systémové požadavky pro spuštění editoru

- Windows XP nebo Windows Vista
- grafická karta s podporou shaderů
- Před spuštěním editoru může být požadována instalace nebo upgrade volně dostupných grafických nebo jiných knihoven třetí stran. (např.: DirectX, JRE, Ogre, Java3D)

Splnění systémových požadavků pro spuštění editoru

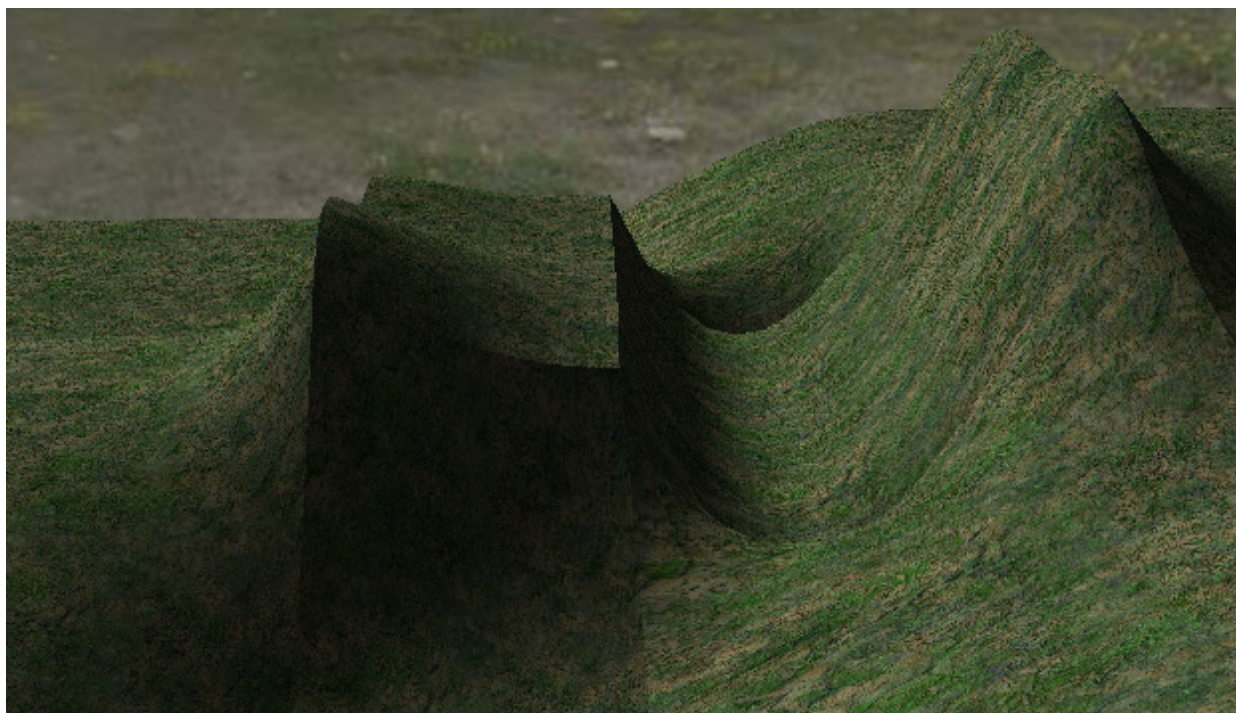
- Systémové požadavky byly splněné nad plán. Vytvořený program jde spustit nejen pod operačními systémy Windows XP a Windows Vista, ale také pod operačními systémy Linux a Solaris. Hlavní důraz na kompatibilitu a testování byl kladen na platformy Windows XP a Windows Vista.
- Před spuštěním editoru není ve většině případů potřeba nic instalovat. Jen v některých případech je třeba doinstalovat Java Runtime Environment nebo knihovnu Java3D.

10.2. Ukázky vymodelovaných terénů

Na závěr práce uvádím několik ukázek vymodelovaného terénu z programu Editor terénů. Na obrázcích 30 a 32 je vidět terén bez ostrých hran. Obrázek 31 demonstruje terén, který obsahuje ostré hrany. Program Editor terénů umožňuje nastavit stupeň spojitosti mezi pláty a tím dosáhnout ostrých i neostrých hran.



Obrázek 30: Ukázka vymodelovaného terénu 1



Obrázek 31: Ukázka vymodelovaného terénu 2



Obrázek 32: Ukázka vymodelovaného terénu 3

Seznam rovnic

| | |
|--|----|
| Rovnice 1: Výpočet normálového vektoru v trojrozměrném případě..... | 12 |
| Rovnice 2: Explicitní vyjádření rovnice kružnice se středem v počátku a poloměrem R..... | 13 |
| Rovnice 3: Výpočet tečného vektoru z explicitního tvaru rovnice křivky ve 2D případě..... | 13 |
| Rovnice 4: Výpočet tečného vektoru z explicitního tvaru rovnice křivky ve 3D případě..... | 13 |
| Rovnice 5: Parametrické vyjádření rovnice kružnice se středem v počátku a poloměrem R.. | 14 |
| Rovnice 6: Výpočet tečny ve 2D případě..... | 14 |
| Rovnice 7: Výpočet tečny ve 3D případě..... | 14 |
| Rovnice 8: Implicitní vyjádření rovnice kružnice se středem v počátku a poloměrem R | 15 |
| Rovnice 9: Výpočet tečného vektoru z implicitního tvaru rovnice křivky ve 2D případě | 15 |
| Rovnice 10: Podmínka geometrické spojitosti řádu n..... | 17 |
| Rovnice 11: Generování souřadnic textury z polohy | 23 |
| Rovnice 12: Generování souřadnic textury z parametru | 23 |

Seznam obrázků

| | |
|---|----|
| Obrázek 1: Nástroj pro 3D modelování 3D Studio Max | 8 |
| Obrázek 2: Nástroj pro 3D modelování Blender | 9 |
| Obrázek 3: Počítačová hra World of Warcraft | 10 |
| Obrázek 4: Virtuální svět Second Life | 11 |
| Obrázek 5: Virtuální svět Lively | 11 |
| Obrázek 6: Vizuální rozdíl mezi parametrickými třídami spojitosti | 16 |
| Obrázek 7: Parametrická a geometrická spojitost. | 17 |
| Obrázek 8: Spojení čtyř Bézierových plátů s C1 spojitostí..... | 24 |
| Obrázek 9: Spojení čtyř Bézierových plátů s G1 spojitostí | 24 |
| Obrázek 10: Ukázka kódu používajícího DirectX | 26 |
| Obrázek 11: Program Editor terénů | 30 |
| Obrázek 12: Hlavní okno Editoru terénů..... | 32 |
| Obrázek 13: Bézierův plát v Editoru terénů | 34 |
| Obrázek 14: Sousedící Bézierovy pláty v Editoru terénů | 35 |
| Obrázek 15: Řídící body pro hraniční Bézierovu křivku | 36 |
| Obrázek 16: Záložka Terrain | 37 |
| Obrázek 17: Dialogové okno pro vytvoření nového terénu | 37 |
| Obrázek 18: Dialogové okno pro otevření souboru s terénem..... | 38 |
| Obrázek 19: Dialogové okno pro uložení souboru s terénem | 38 |
| Obrázek 20: Záložka Tools | 39 |
| Obrázek 21: Původní stav - terén s C1 spojitostí..... | 39 |
| Obrázek 22: Terén se sníženou spojitostí | 40 |
| Obrázek 23: Ostrý zlom vzniklý kvůli snížení třídy spojitosti | 40 |
| Obrázek 24: Záložka Actions | 41 |
| Obrázek 25: Drátěný model terénu | 42 |
| Obrázek 26: Záložka Mass move | 42 |
| Obrázek 27: Režim Quadratic..... | 43 |
| Obrázek 28: Režim Linear..... | 44 |
| Obrázek 29: Režim Constant | 44 |
| Obrázek 30: Ukázka vymodelovaného terénu 1 | 47 |
| Obrázek 31: Ukázka vymodelovaného terénu 2 | 48 |
| Obrázek 32: Ukázka vymodelovaného terénu 3 | 48 |

Použitá literatura

1. **Windrow, Richard.** *Terrain Modelling.* : Osprey Publishing, 2001.
2. **Polack, Trent.** *Focus On 3D Terrain Programming.* : Course Technology PTR, 2002.
3. Autodesk 3D Studio Max. *Autodesk.* [Online]
<http://usa.autodesk.com/adsk/servlet/index?id=5659302&siteID=123112>.
4. 3D Studio Max. *3D Shop.cz.* [Online] <http://www.3dshop.cz/software/3d-studio-max-9>.
5. *Blender.* [Online] www.blender.org.
6. **Roosendaal, Ton a Selleri, Stefano.** *The Official Blender 2.3 Guide.* : No Starch Press, 2005.
7. **Whitehead, James, McLemore, Bryan a Orlando, Matthew.** *World of Warcraft Programming: A Guide and Reference for Creating WoW Addons.*
8. **Mansfield, Richard.** *How to Do Everything with Second Life.* : McGraw-Hill Osborne Media, 2007.
9. **Žára, Jiří, a další.** *Moderní počítačová grafika.* 2004.
10. **Gillies, Duncan Fyfe.** Spline Curves and Surfaces. [Online] květen 2008.
<http://www.doc.ic.ac.uk/~dfg/AndysSplineTutorial/>.
11. **Alexandr, Lubomír.** Křivky. [Online] květen 2008.
http://lubovo.misto.cz/_MAIL_/curves/krivky.html.
12. **Skála, Václav.** Křivky v počítačové grafice. [Online] květen 2008.
<http://herakles.zcu.cz/education/ZPG/cviceni.php?no=8>.
13. **Finney, Kenneth C.** *3D Game Programming All in One.* : Course Technology PTR, 2006.
14. **Sherrod, Allen.** *Game Graphics Programming.* : Charles River Media, 2008.
15. **Ahearn, Luke.** *3D Game Textures.* : Focal Press, 2006.
16. **Summers, Dennis.** *Texturing: Concepts and Techniques.* : Charles River Media, 2004.
17. OpenGL - The Industry Standard for High Performance Graphics. [Online] Květen 2008.
<http://www.opengl.org/>.
18. **Board, OpenGL Architecture Review.** *OpenGL Programming Guide: The Official Guide to Learning OpenGL.* : Addison-Wesley Professional, 2007.

19. **Martz, Paul.** *OpenGL Distilled.* : Addison-Wesley Professional, 2006.
20. Quick Example Code Snippets. *GameDev.* [Online]
<http://www.gamedev.net/reference/articles/article578.asp>.
21. Learn DirectX. *MSDN.* [Online] Květen 2008. [http://msdn.microsoft.com/cs-cz/directx/bb896684\(en-us\).aspx](http://msdn.microsoft.com/cs-cz/directx/bb896684(en-us).aspx).
22. **Snook, Greg.** *Real-Time 3D Terrain Engines Using C++ and DirectX 9.* : Charles River Media, 2003.
23. **Zhang, Hong a Liang, Daniel.** *Computer Graphics Using Java 2D and 3D.* : Prentice Hall, 2006.
24. **Davison, Andrew.** *Pro Java 6 3D Game Development.* : Apress , 2007.
25. —. *Killer Game Programming in Java.* : O'Reilly Media, Inc., 2005.
26. **Selman, Daniel.** *Java 3D Programming.* : Manning Publications, 2002.

Obsah přiloženého CD

Na disku CD, který je součástí diplomové práce naleznete:

1. **Text diplomové práce** ve formátu PDF
2. **Program Editor terénů** ve spustitelné podobě
3. **Zdrojové kódy** programu Editor terénů
4. **Ukázky vymodelovaných terénů** ve formátu XML
5. **Běhové prostředí pro Javu** - Java Runtime Environment
6. **Knihovnu Java3D** pro Windows