

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

Počítačové modelování a zobrazení 3D terénu

Miroslav Dráb

©ČZU 2012 v Praze

Čestné prohlášení

Prohlašuji, že svoji diplomovou práci "Počítačové modelování a zobrazení 3D terénu" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 3.4.2012

.....

Poděkování

Rád bych poděkoval Ing. Vynikarové, Ph.D. za vedení a cenné rady při tvorbě diplomové práce.

Počítačové modelování a zobrazení 3D terénu

Computer simulation and representation of 3D terrain

Souhrn

Práce se zabývá metodami získávání a reprezentace dat pro modelování terénu. Jsou popsány základní techniky počítačového modelování, získávání terénních dat, datové struktury pro reprezentaci terénu a metody urychlující zobrazování modelů. V praktické části je implementována jednoduchá aplikace pro generování terénu. Aplikace umožňuje náhodně vygenerovat model terénu, dále ho upravovat a výsledek zobrazit v podobě 3D modelu. Do modelu je možné přidat vodní hladinu a skybox, což je objekt, který obklopuje terén a jeho stěny jsou pokryty texturou okolního prostředí. Pro přirozenější vzhled výsledného terénu aplikace podporuje funkce pro vyhlazování povrchu terénu.

Summary

This paper studies methods of acquiring and representation of data for terrain modelling. There are described methods of computer modelling, terrain data acquisition, terrain data structures and methods for models displaying. Practical section describes implementation of simple terrain generator application. Application implements functions for terrain generation and terrain modification. The final terrain model can be rendered as a 3D model. Model can be enhanced by water layer and skybox. Skybox is an object whose faces are covered by textures of the surrounding environment. For better natural look the application also implements terrain surface smoothing function.

Klíčová slova: 3D počítačová grafika, modelování, OpenGL, terén, úpravy terénu

Keywords: 3D computer graphics, modelling, OpenGL, terrain, terrain modifications

Obsah

1	Úvod	1
2	Cíl práce a metodika	2
3	Problematika počítačového modelování	3
3.1	Modelování a reprezentace těles	3
3.1.1	Mnohoúhelníkové povrchy, polygony	3
3.1.2	Zakřivené povrchy a křivky	5
3.1.2.1	Kvadriky	5
3.1.2.2	Parametrické křivky	5
3.1.2.3	Aproximační křivky	6
3.1.3	Procedurální metody	6
3.1.3.1	Částicové systémy	7
3.1.4	Hraniční reprezentace	8
3.1.4.1	Hranová reprezentace	8
3.1.4.2	Plošková reprezentace	8
3.1.4.3	Bodová reprezentace	9
3.1.5	Konstruktivní geometrie těles (CSG)	9
3.2	Zdroje dat	10
3.2.1	Kartografické zdroje	10
3.2.2	Letecké a kosmické snímkování	11
3.2.3	Fotogrammetrie	11
3.2.4	Letecké laserové skenování	11
3.2.5	Umělé generování krajin	12
3.2.5.1	Metoda přesouvání středního bodu	12
3.2.5.2	Metoda náhodných poruch	13

3.2.5.3	Usazování částic	15
3.3	Digitální model terénu	15
3.3.1	Datové modely využívané v digitálních modelech terénu	16
3.3.1.1	Vrstevnicový model	16
3.3.1.2	Point-based modelování	16
3.3.1.3	Pravidelná mřížka	17
3.3.1.4	Nepřavidelná trojúhelníková síť	18
3.3.1.5	Hybridní přístup	18
3.4	Interpolace dat	19
3.4.1	Exact fitting	20
3.4.1.1	Lineární interpolace	20
3.4.1.2	Bilineární interpolace	21
3.4.1.3	Metoda nejbližšího souseda	22
3.4.2	Best fitting	22
3.4.3	Globální interpolační metody	23
3.4.3.1	Analýza trendu povrchu	23
3.5	Zobrazení	24
3.5.1	Snížení náročnosti renderování scény	24
3.5.1.1	Graf scény	25
3.5.1.2	BSP stromy	26
3.5.1.3	Čtyřstromy a osmístromy	27
3.5.1.4	Úroveň podrobností	27
3.5.2	Skybox	28
3.6	Využití v praxi	29
3.6.1	Stavebnictví	29
3.6.1.1	Navrhování železnic a silnic	29
3.6.1.2	Modelování odtokových systémů	31
3.6.2	Výzkum planety Země	31
3.6.2.1	Povětrnostní modely	31
3.6.2.2	Modelování rizika vzniku požárů	31
3.6.3	Plánování a správa surovin	32
3.6.4	Vojenství	32
3.7	Použité technologie	32
3.8	Grafické API OpenGL	33

3.8.1	Míchání barev (blending)	33
4	Implementace programu pro generování a zobrazení 3D terénu	35
4.1	Popis uživatelského rozhraní	35
4.2	Editor terénů	40
4.2.1	Náhodný generátor terénu	40
4.2.1.1	Přesouvání středního bodu	40
4.2.1.2	Metoda náhodných poruch	43
4.2.1.3	Usazování částic	47
4.2.2	Vyhlazování terénu	50
4.2.3	Ruční úpravy terénu	50
4.2.3.1	Metoda MakeHillFlat	51
4.2.3.2	Metoda MakeHillRounded	51
4.2.4	Skybox	54
4.2.5	Vodní plochy	54
4.3	Implementace	58
4.3.1	Objektový model	58
4.3.1.1	TerrainGenerator	58
4.3.1.2	SkyBox	59
4.3.1.3	Water	59
4.3.1.4	Terrain	59
5	Závěr	61
6	Slovníček pojmů	62
	Literatura	64

Kapitola 1

Úvod

Počítačové modelování je v současné době významnou disciplínou, jejíž poznatky nacházejí uplatnění v mnoha oborech lidské činnosti. Počítačové modely jsou využívány v netechnických i technických oborech. V netechnických oborech se počítačové modely využívají např. při návrhu předmětů denní potřeby, jejich tvaru a textury, při rekonstrukci rentgenových snímků nebo také v kinematografii a počítačovém výtvarném umění. V technických oborech nacházejí využití ve strojírenském průmyslu (letecký, loďařský, automobilový) nebo při výrobě různých tvarově složitých součástek (lopatek...). Do této kategorie spadají i geografické informační systémy, které se zabývají, mimo jiné, také digitálním modelováním terénu.

Vytvořené modely jsou využívány pro simulace různých jevů, např. v biologii pro studium šíření látek lidským organismem, hledání ideální struktury enzymů a proteinů pro léčbu chorob, v leteckém průmyslu se využívají pro nalezení tvaru letounu s nejmenším odporem vzduchu apod.

Digitální modely terénu a jejich úpravy jsou využívány pro studium a simulaci jevů probíhajících v ekosystému Země, nalézají uplatnění při tvorbě povětrnostních modelů, modelování rizika vzniku požárů a jejich šíření. Modely se využívají také ve stavebnictví při stavbě budov nebo komunikačních tras (silnice, železnice). Velký význam mají modely terénů pro armádu, která je využívá jako data pro simulátory, navádění střel aj.

Kapitola 2

Cíl práce a metodika

Tato diplomová práce se zaměřuje na algoritmy pro generování modelů terénu a jejich zobrazení. V teoretické části práce jsou popsány metody, které se využívají v souvislosti s digitálními modely terénu. V části „Modelování a reprezentace těles“ jsou popsány obecné zákonitosti 3D počítačové grafiky a renderování scén. Tato část se zaměřuje na různé způsoby modelování a reprezentace objektů v počítačové grafice. V části „Zdroje dat“ jsou popsány některé způsoby získávání dat pro tvorbu modelů terénu. Jsou popsány metody pro získávání reálných i uměle generovaných dat. V kapitole „Digitální model terénu“ jsou popsány různé datové modely pro uchování a manipulaci s daty pro reprezentaci terénu. Následující kapitola, „Interpolace dat“, se zabývá interpolací a odvozováním nových informací z již naměřených dat. Závěrečné kapitoly teoretické části práce se zaměřují na vizualizaci modelů terénu a jejich využití v praxi. V poslední kapitole teoretické části jsou informace o grafickém API OpenGL, které je využito pro splnění cíle praktické části práce.

Cílem praktické části práce je implementace programu pro generování a renderování modelů terénu.

Při naplňování vytyčených cílů autor čerpá z odborné literatury, internetových zdrojů a ze svých zkušeností s počítačovou grafikou, modelováním a grafickým API OpenGL.

Kapitola 3

Problematika počítačového modelování

3.1 Modelování a reprezentace těles

Počítačové modelování se zabývá možnostmi popisu těles zobrazovaných pomocí počítače. Zobrazovaná scéna může obsahovat různé druhy objektů: stromy, rostliny, mraky, vodu atd. Protože modelované objekty jsou dosti různorodé *”žádná metoda, kterou lze použít pro jejich popis, nebude zachycovat všechny vlastnosti různých materiálů”*, a proto je nutné *”použít reprezentace, které přesně modelují vlastnosti”* konkrétních *”objektů”*[5, kapitola 10, s. 305].

3.1.1 Mnohoúhelníkové povrchy, polygony

Polygony poskytují přesný popis pro jednoduché euklidovské objekty jako jsou mnohostěny a elipsoidy. Jde o nejpoužívanější techniku popisu 3D objektu metodou hraniční reprezentace (Hraniční reprezentace viz kapitola 3.1.4). Každý objekt je popsán množinou mnohoúhelníků, které tvoří jeho hranici. Výhodou tohoto přístupu je *”zjednodušení a zrychlení vykreslování povrchu a objektu, neboť všechny povrchy jsou popsány lineárními rovnicemi”*[5, kapitola 10.1, s. 305]. Pokud model není možné popsat přímo, je nutné jej převést do vhodné formy. Tento proces se nazývá tessellace.

Pro uložení informací o mnohoúhelnících se používají následující datové struktury[5]:

- tabulky mnohoúhelníků – povrch je vyjádřen *”množinou souřadnic vrcholů a souvisejících atributů”*, přičemž lze rozlišovat dva druhy tabulek: tabulku pro popis geometrických dat, *”která obsahuje souřadnice vrcholu a informace týkající se prostorové orientace”*, a tabulku vlastností objektu, mezi které patří *”úroveň průhlednosti, odrazivost povrchu a charakteristika textury”*[5, kapitola 10.1, s. 306], sklon hran aj. Tato data nemusí být uložena pouze v jedné tabulce. Z hlediska práce s daty je nejpohodlnější pro jejich uložení použít tři tabulky: tabulku vrcholů, tabulku hran a tabulku mnohoúhelníků. Tato reprezentace poskytuje nejvíce informací o datech a lze z ní jednoduše zobrazit drátěný model objektu. Lze využít i reprezentace o méně tabulkách: tabulku vrcholů a tabulku polygonů, kde však může dojít k *”více násobnému vykreslení některých hran”*, případně pouze tabulku polygonů, při jejímž použití dochází k *”duplicitě informací o souřadnicích vrcholů”* neboť informace o *”souřadnicích jsou uloženy zvlášť pro každý vrchol každého polygonu”*[5, kapitola 10.1, s. 306]
- rovinné rovnice – vhodné pro vyjádření prostorové orientace objektu. Každý polygon je vyjádřen rovnicí [5, kapitola 10-1, s. 308]:

$$Ax + By + Cz + D = 0$$

kde A, B, C jsou složky normálového vektoru a D určuje vzdálenost roviny od počátku. Všechny koeficienty lze vypočítat ze soustavy rovnic [5, kapitola 10-1, s. 308]:

$$\begin{aligned} A &= y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2) \\ B &= z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2) \\ C &= y_1(y_2 - y_3) + y_2(y_3 - y_1) + y_3(y_1 - y_2) \\ D &= -x_1(y_2z_3 - y_3z_2) - x_2(y_3z_1 - y_1z_3) - x_3(y_1z_2 - y_2z_1) \end{aligned}$$

U tohoto způsobu uložení dat je důležitá směrnice normálového vektoru, protože v případě, že modelujeme *”polygonové povrchy, které uzavírají vnitřek objektu, je potřeba rozlišit mezi dvěma stranami polygonu”*[5, kapitola 10.1, s. 108]: vnitřní a vnější. Pokud jsou vrcholy povrchu zadány proti směru hodinových ručiček, míří normála tohoto povrchu směrem ven z objektu.

3.1.2 Zakřivené povrchy a křivky

Složitější struktura objektů může být popsána modely se zakřiveným povrchem. Křivky se využívají pro popis průběhu funkcí a grafů. Povrchy většinou nebývají popsány funkcemi, ale jsou *"rozloženy na polygony aproximující povrch"*[5, kapitola 10.2, s. 310] objektu. Místo polygonů se často používají pásy trojúhelníků, neboť u polygonů *"se čtyřmi nebo více vrcholy nemusí všechny vrcholy ležet v jedné rovině"*[5, kapitola 10.2, s. 310]

Pro popis zakřivených povrchů se často využívá objektů s kvadratickým povrchem, což jsou objekty *"které jsou popsány rovnicemi druhého stupně"*[5, kapitola 10.3, s. 310]. Někdy se těmto objektům říká kvadriky (quadrics). Mezi kvadriky jsou řazeny koule, elipsoidy, válce, paraboloidy a hyperboloidy. Spline křivky a konstrukční techniky se používají při *"návrhu křídel letadel, výstroje, a jiných inženýrských struktur se zakřivenými povrchy"*[5, kapitola 10, s. 305], dále pro *"definici fontů, určování dráhy pohyblivých se objektů"*[1, kapitola 5, s. 177] a jiné. Spline křivka je *"po částech polynomiální křivka"*[1, kapitola 5.1, s. 179], která *"má ve všech bodech spojitě derivace"*[1, kapitola 5.4.4, s. 192]

3.1.2.1 Kvadriky

Jedná se o různé rotační objekty a objekty popsatelné kvadratickými rovnicemi. Podrobnější popis lze nalézt např. v [5, kapitola 10.3, s. 310]. V počítačové grafice jsou tyto objekty převedeny na síť trojúhelníků. Modelovací programy a API pro práci s grafikou mají pro kvadriky vytvořenu sadu funkcí.

3.1.2.2 Parametrické křivky

Pro vyjádření křivek v počítačové grafice se nejčastěji používá jejich parametrické vyjádření [1, kapitola 5.1, s. 178]:

$$x = x(t), y = y(t), z = z(t)$$

kde parametr t představuje časovou proměnnou a $t \in \langle 0, 1 \rangle$. Výhodou tohoto vyjádření je *"závislost souřadnic křivky na jediném parametru, jehož fyzikální interpretací je čas"*[1, kapitola 5.1, s. 178], což umožňuje modelovat i křivky, které *"procházejí vícekrát stejnými body v prostoru"*[1, kapitola 5.1, s. 179].

Za základní třídu parametrických křivek jsou v počítačové grafice považovány polynomiální křivky. Matematicky je lze vyjádřit rovnicí [1, kapitola 5.2, s. 181]:

$$Q_n(t) = a_0 + a_1t + \dots + a_nt^n$$

Nejčastěji jsou využívány křivky třetího stupně, nazývané kubiky, neboť poskytují "širokou škálu tvarů, jejich výpočet bývá nenáročný" [1, kapitola 5.2, s. 181] a lze s nimi snadno manipulovat. Při tvarování křivky do požadovaného tvaru je její vyjádření rovnicemi nepraktické, protože z nich nelze snadno odhadnout její tvar. Proto jsou při modelování křivek používány "viditelné prvky, jako jsou řídicí body, směry a tečné vektory, zakřivení"[1, kapitola 5.2, s. 181], které umožňují měnit parametry křivky a sledovat nastalé změny.

3.1.2.3 Aproximační křivky

Mezi patrně nejpoužívanější aproximační křivky, které jsou využívány pro modelování ve 2D i 3D prostoru, ale např. i pro definici písma (fontů), patří Beziérovky křivky. Beziérovka n -tého stupně je určena rovnicí [1, kapitola 5.4, s. 185]:

$$Q(t) = \sum_{i=0}^n P_i B_i^n(t)$$

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}; t \in \langle 0, 1 \rangle; i = 0, 1, \dots, n$$

Pro tento vztah platí, že $\binom{n}{0} = 1$ a $0^0 = 1$

Mezi vlastnosti Beziérových křivek patří, že "při změně polohy jediného řídicího bodu dojde ke změně tvaru celé křivky"[1, kapitola 5.4.1, s. 186]. Tento nedostatek, který může komplikovat návrh složitých tvarů křivek, je řešen složením složité křivky z více na sebe navazujících jednodušších křivek.

3.1.3 Procedurální metody

Procedurální metody se používají pro "přesnou reprezentaci pro mraky, trsy trávy a další přírodní objekty"[5, kapitola 10, s. 305]. Jde o objekty, jejichž tvar je obtížně popsitelný rovnicemi a jejich soustavami. Pro popis jejich vlastností jsou využity nej-

různější algoritmy. Procedurální metody lze dle [1] dělit mimo jiné na:

- algoritmy vycházející z gramatik – L-systémy, využívány pro generování rostlin.
- fraktální geometrie – algoritmy pro generování krajin, hor, stromů.
- částicové systémy – generování explozí, fontán, dýmu aj.

Dále budou podrobněji popsány částicové systémy.

3.1.3.1 Částicové systémy

Některé objekty nelze snadno popsat soustavou rovnic. Jedná se o objekty *”nepravidelného tvaru”*, který se *”mění v průběhu času”*[5, kapitola 10-20, s. 390]. Jde o objekty typu ohňostroj, vodotrysk, trsy trávy, plyn, ale i hejna ptáků nebo ryb. Místo složitého popisování rovnicemi, je modelovaný objekt nahrazen množstvím malých objektů, nazývaných částice, jejichž vlastnosti jsou proměnlivé v čase. Částice mohou být popsány vlastnostmi, které určují jejich barvu, směr pohybu, rotaci, tvar, stáří apod. Některé vlastnosti mohou být závislé nejen na čase (stáří částice), ale mohou na ně působit i síly okolního prostředí, např. gravitační síla ovlivňuje směr pohybu částice. Částice mohou *”v jistém okamžiku a místě vznikat, po určité době života mizí, mohou emitovat další částice, srážet se mezi sebou a s okolními objekty”*[1, kapitola 8.3, s. 288].

Každá částice je reprezentována malým, a většinou i jednoduchým, geometrickým objektem typu koule nebo krychle, bodem, ale i složitějšími objekty jako sněhová vločka nebo kapka. Každá částice má určenu délku života, po jejímž uplynutí zaniká. O vznik nových částic se stará generátor (emitor), který do systému přidává nové částice a určuje jejich počáteční vlastnosti.

Podle [1, kapitola 8.3, s. 289] může jedna iterace simulace částicového systému vypadat takto:

1. aktualizace stavu existujících částic na základě jejich atributů.
2. tvorba nových částic. Částice vznikají buď v určité oblasti, nebo jako potomci jiné částice.
3. přiřazení počátečních vlastností každé z nových částic.
4. odstranění částic, které přesáhly dobu svého života.

5. zobrazení celého systému.

Obrázek 3.1 ukazuje jednu z možných implementací částicového systému.



Obrázek 3.1: Ukázka částicového systému (převzato z <http://nehe.gamedev.net>)

3.1.4 Hraniční reprezentace

Jedná se o nejběžněji používanou reprezentaci 3D objektů. Objekt je popsán množinou primitiv, která oddělují jeho vnější a vnitřní oblast.

3.1.4.1 Hranová reprezentace

Jedná se o nejstarší používanou metodu. Geometrie modelu je popsána seznamem vrcholů a seznamem hran. Tímto uložením je dosaženo *”úsporné struktury, která však obsahuje minimum topologických informací”*, a proto výsledný model *”nelze jednoznačně interpretovat”* [1, kapitola 6.2.3, s. 243]. Jeden a ten samý model může být, kvůli nepřítomnosti informace o plochách, interpretován více způsoby.

3.1.4.2 Plošková reprezentace

Jde o rozšíření předchozí struktury. K seznamu vrcholů a hran se přidá seznam ploch (stěn), který obsahuje ukazatele do seznamu vrcholů. Plochy je nejlepší ukládat jako trojúhelníky, neboť *”topologii jejich sítě lze snadno zaznamenat, např. do pole, jehož*

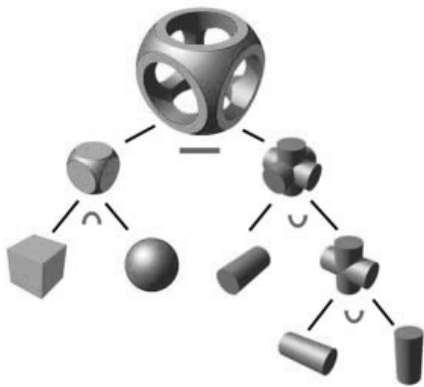
každá položka pevné délky obsahuje právě tři ukazatele do seznamu vrcholů”[1, kapitola 6.2.4, s. 244]. Z pořadí uložených vrcholů lze určit směr normálového vektoru stěny tělesa tj., zda je vnitřní nebo vnější. Tato struktura je vhodná pro objekty, u kterých je potřeba zohlednit viditelnost jednotlivých částí při jejich vykreslování.

3.1.4.3 Bodová reprezentace

Objekt je vyjádřen množinou povrchových bodů. Body jsou většinou získány digitálním snímáním jeho povrchu. Každý bod *”představuje určitou část povrchu dané velikosti a nese informaci o své poloze, normálovém vektoru, barvě, případně dalších vlastnostech týkajících se odrazu světla”*[1, kapitola 6.2.6, s. 246]. Bodová reprezentace poskytuje vysokou přesnost digitalizovaného modelu, ale je značně paměťově náročná.

3.1.5 Konstruktivní geometrie těles (CSG)

Konstruktivní geometrie těles je založena na *”analytickém popisu těles jejich objemem, tj. podmnožinou trojrozměrného prostoru ležícího uvnitř tělesa”*[9]. Popisuje těleso množinou operací, které používá konstruktér při vytváření skutečného objektu. Mezi operace se řadí jednoduché booleovské operace: sjednocení, průnik, rozdíl. Jako výchozí těleso se používá nějaký jednoduchý objekt: kvádř, koule, válec, kužel, toroid. Sled operací vedoucí ke vzniku výsledného objektu je reprezentován *”stromovou strukturou (CSG stromem) uchováující historií dílčích konstrukčních kroků”*[1, kapitola 6.3, s. 246].



Obrázek 3.2: CSG strom (převzato z <http://cad-developers.livejournal.com/>)

Obrázek 3.2 ukazuje, jak může CSG strom vypadat. Ve vnitřních uzlech jsou uloženy informace o operacích, údaje o CSG primitivech jsou uloženy v listech. Tato metoda je určena pro uložení informací o geometrii tělesa, pro jeho vykreslení je vhodnější CSG strom převést na některou z dříve zmíněných reprezentací.

3.2 Zdroje dat

Většina digitálních modelů terénu reprezentuje nějaký skutečně existující kus terénu. Aby mohly tyto modely vzniknout, musí nejdříve být k dispozici data, ze kterých budou modely vytvořeny. Podle pravidelnosti rozmístění informace o výšce terénu lze data dělit na:

- pravidelně distribuovaná - data jsou umístěna v pravidelné mřížce.
- nepravidelně distribuovaná - data jsou měřena pouze v místech, kde zachycují nějaký důležitý rys na modelovaném terénu.

V dnešní době existuje mnoho způsobů, jak tato data získat.

3.2.1 Kartografické zdroje

Jde o převod existujících map do digitální podoby. Podle [7] existuje několik možností získání dat z kartografických map. Jedná se o:

- ruční sledování čar (Manual line following) – materiál určený k digitalizaci se položí na digitalizační stůl. Operátor poté kurzorem kopíruje objekty, jejichž pozici chce zdigitalizovat. Poloha objektu je zaznamenána buďto při kliknutí, po uplynutí určitého časového intervalu nebo poté, co kurzor překoná určitou vzdálenost. V praxi se jedná o nejpoužívanější metodu.
- automatické sledování čar (Automatic line following) – zařízení po nastavení na počátek čáry (kontury), kterou má zaznamenat, sleduje její trasu a automaticky zaznamenává údaje. Zásah operátora je nutný pro rozhodnutí sporných případů nebo zaznamenávání doplňujících údajů např. o výšce atd.

- skenování (automatické nebo manuální) – převedení mapy pomocí skeneru do digitální podoby. Pro kartografické účely jsou použity skenery s vysokým rozlišením. Po naskenování jsou data většinou převedena do vektorové podoby, ať již manuálně nebo automaticky, čímž je snížena redundance a jejich objem.

Po skončení digitalizace je nutné provést konverzi souřadnic, neboť *”data získaná digitalizací jsou uložena v souřadnicovém systému digitalizačního zařízení”* a musí být *”převedena na geodetický souřadnicový systém”*[7, kapitola 3.5.1, s. 57].

3.2.2 Letecké a kosmické snímkování

Další možností získávání informací o terénu jsou fotografie pořízené za pomoci letadel nebo družic. Jedná se o nejefektivnější způsob získání dat o terénu. Fotografie jsou většinou pořizovány v analogové formě a v případě potřeby převedeny na digitální formu (např. skenováním)[7, kapitola 3.1.2, s. 33]. Jinou možností je pořizovat fotografie přímo v digitální podobě. Další, méně používanou, možností je využít obrazy pořízené družicemi. Takto pořízené obrazy jsou *”využívány pro tvorbu digitálních modelů terénu velkých oblastí v malém měřítku”*[7, kapitola 3.1.2, s.33]. S nástupem družic pořizujících obrazy ve vysokém rozlišení (např. IKONOS) budou takto pořízená data pravděpodobně využívána ve větší míře. V tomto případě jsou data získána pasivním způsobem.

3.2.3 Fotogrammetrie

Fotogrammetrie je disciplína zabývající se rekonstrukcí 3D objektu z jednoho páru stereo obrazů. Stereo pár jsou *”dva obrazy stejné scény pořízené z mírně odlišných pozic, takže se do určité míry překrývají”*[7, kapitola 3.2.2, s. 37]. 3D model je vytvořen z těchto překrývajících se částí. Bez překrytí by nebylo možné jej vytvořit. Míra překrytí je poměrně velká, při leteckém snímkování dochází k *”60% překryvu ve směru letu a k 30% překryvu sousedních pásů”*[7, kapitola 3.2.2, s. 37].

3.2.4 Letecké laserové skenování

Během 60. a 70. let se začal pro dálkový průzkum využívat laser (ALS – airborne laser scanning). ALS je využíván, pokud fotogrammetrii nelze použít, nebo je její využití příliš nákladné. Jde například o *”zjištění výšky terénu v zalesněných oblastech”*[7,

kapitola 3.4, s. 51] nebo *”získání prostorových dat městských oblastí”*[7, kapitola 3.4, s. 51]. LIDAR (Light Detection And Ranging) je aktivní systém, který vysílá *”elektromagnetickou energii a zaznamenává energii odraženou zpátky od povrchu terénu a objektů na něm”*[7, kapitola 3.4.1, s. 53]. Typ objektu, od kterého se paprsek odrazil, lze určit podle intenzity vrátivšího se signálu. Výsledkem měření systémem LIDAR je shluk trojrozměrných bodů. Přesnost měření závisí na přesnosti dat dodávaných z GPS modulu. Nepřesnosti měření mohou být způsobeny *”odchytkami v určení pozice a orientace letounu, úhlu paprsku, atmosférických vlivech”*[8, kapitola 2.7, s. 46] a jinými vlivy. Tyto vlivy způsobují, že LIDAR dosahuje *”vertikální přesnosti od 10 do 20 cm a horizontální přesnosti do 1 m”*[8, kapitola 2.7, s. 46]. Než jsou naměřené body použity pro vytvoření modelu projdou procesem filtrování a klasifikací. Filtrování je zaměřeno na *”odstranění nechtěných měření a nalezení povrchu terénu ve směsi bodů země a vegetace”*[7, kapitola 3.4.2, s. 55]. Úkolem klasifikace je *”nalezení určitých geometrických nebo statistických struktur”*[7, kapitola 3.4.2, s. 55], jako jsou například budovy. Odstraněním objektů jako jsou budovy, stromy atd. vznikne výsledný model terénu.

3.2.5 Umělé generování krajín

Mimo převádění skutečných terénů do digitální podoby je také možné celý terén vygenerovat uměle. Uměle vytvořené terény nacházejí uplatnění v mnoha odvětvích, *”z nichž patrně nejpodstatnější jsou letecké simulátory a videohry”*[1, kapitola 8.2.2, s. 282]. Samotné generování probíhá ve dvou krocích:

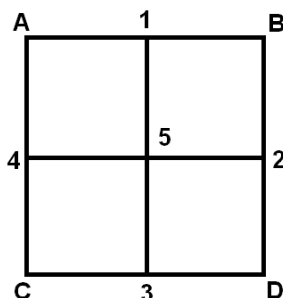
- vygenerování hrubého obrysu terénu,
- úprava získaných dat.

Pro získání hrubého obrysu terénu se využívá metod fraktální geometrie. Mezi metody, které využívají tuto oblast matematiky, jsou řazeny: metoda přesouvání středního bodu (midpoint displacement, diamond-square algorithm), metoda náhodných poruch (fault formation) a metoda usazování částic (particle deposition).

3.2.5.1 Metoda přesouvání středního bodu

Ve středu každé hrany čtverce je vytvořen bod (body 1,2,3,4 na obr. 3.3). Jeho výška je rovna průměru výšky koncových bodů hrany změněná o náhodnou hodnotu. Poté je

vytvořen bod ležící ve středu čtverce (5). Jeho výška je vypočtena jako průměr výšek bodů ve středech hran (1,2,3,4) ke které je připočtena náhodná hodnota γ (lineární nebo Gaussovské náhodné číslo). γ se může zmenšovat s přibývajícím hloubkou rekurze a lze ji vypočítat např. podle vzorce: $\gamma = \gamma * 2^{-\text{požadovana_hrubost_terenu}}$ [4, kapitola 2, s. 34]. Tento postup se opakuje pro každý ze vzniklých čtverců. V jiné verzi algoritmu je nejdříve vypočten bod 5 a až poté středy hran (1,2,3,4 kde $1 = (A + B + 5)/3 + n$ apod.). [6, s. 3]. Ukáázka terénu generovaného metodou přesouvání středního bodu je na obr. 3.4.



Obrázek 3.3: Princip metody přesouvání středního bodu



Obrázek 3.4: Přesouvání středního bodu (převzato z <http://www.lighthouse3d.com>)

3.2.5.2 Metoda náhodných poruch

Princip metody spočívá v rozdělení terénu na dvě části. Výška bodů v jedné z nich je zvýšena o náhodnou veličinu *Height*. Výška bodů v druhé části je naopak snížena o stejnou hodnotu. Hodnota *Height* se zmenšuje s každou další iterací. Pokud by tomu tak nebylo, vznikl by "*chaotický terén bez kulatých kopců*"[4, kapitola 2, s. 28].

Rovina může být rozdělena přímkou, nebo jiným geometrickým obrazcem. Obrazec většinou rovinu nerozdělí na dvě stejné části, proto je nutné "*měnit periodicky*,

KAPITOLA 3. PROBLEMATIKA POČÍTAČOVÉHO MODELOVÁNÍ

nebo náhodně, avšak se stejnou pravděpodobností, oblasti, jejichž hodnoty se zvyšují a snižují”[1, kapitola 8.1, s. 281]. Hodnota *Height* se může s počtem iterací snižovat. Hodnotu *Height* lze vypočítat ze vzorce:

$$Height = MaxDelta - \frac{(MaxDelta - MinDelta) \cdot CurrentIteration}{Iterations}$$

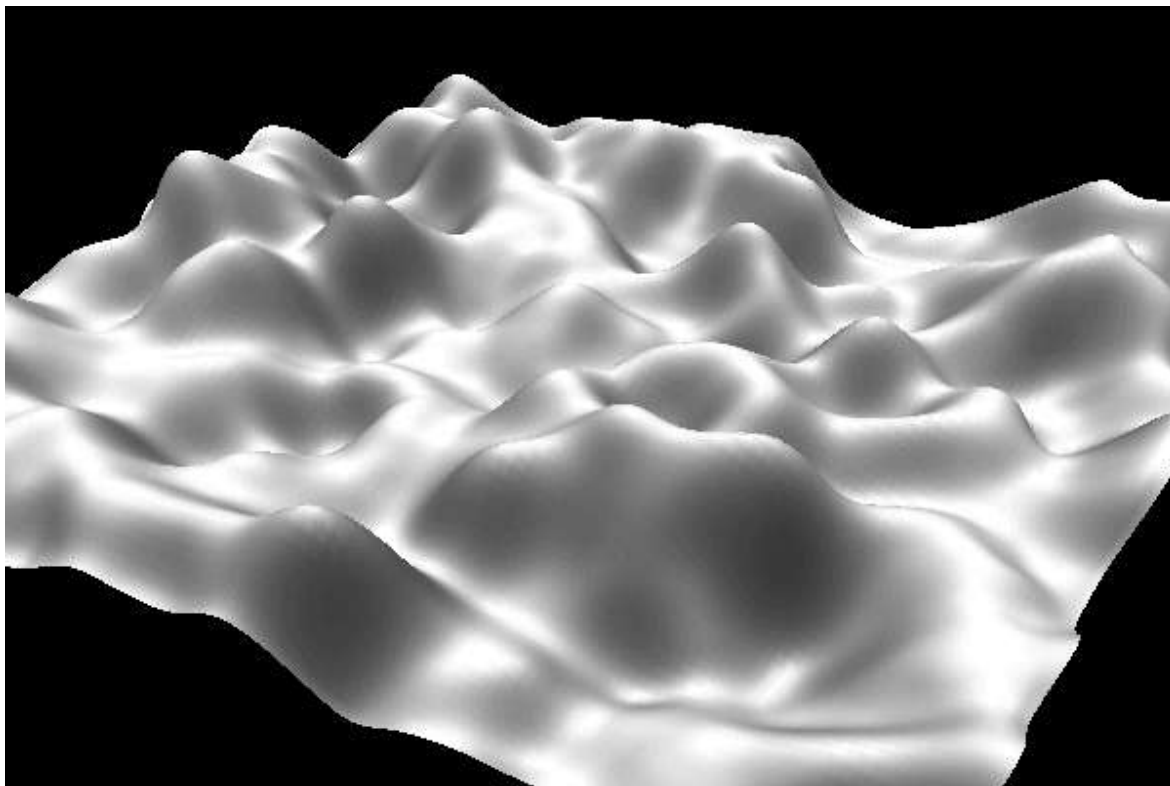
MaxDelta – nejvyšší možná výška terénu

MinDelta – nejnižší možná výška terénu

Iterations – počet iterací algoritmu

CurrentIteration – aktuální počet proběhlých iterací

Model vytvořený touto metodou je zobrazen na obr. 3.5.



Obrázek 3.5: Metoda náhodných poruch (převzato z <http://www.lighthouse3d.com>)

3.2.5.3 Usazování částic

Výsledný terén je získán iterativním opakováním následujících kroků. Na plochu je náhodně umístěna částice tj. výška terénu je v místě dopadu částice zvýšena o k . Částice se poté *"pohybuje směrem dolů, dokud není ve stabilní pozici"*. Za stabilní je považována pozice, ve které je výška terénu v bezprostředním okolí částice nižší, než její aktuální výška. Pokud se v sousedství částice vyskytuje oblast s výškou $n - m$, kde n je současná výška částice a $m \leq n$, částice se na tuto oblast může přesunout.

Při každém pohybu částice je zároveň zvýšena výška terénu v místě její nové polohy o k jednotek, přičemž k se s počtem iterací postupně snižuje. Sousedství částice může být různě velké, čímž lze dosáhnout eroze a různého stupně svažitosti kopců. Odlišného vzhledu terénu lze také dosáhnout úpravou veličiny m .

Modely vzniklé touto metodou jsou zobrazeny na obr. 3.6.



Obrázek 3.6: Usazování částic (převzato z <http://wtomandev.blogspot.com>)

3.3 Digitální model terénu

Matematický (digitální, DTM - digital terrain model) model terénu je množina matematických funkcí, které slouží k reprezentaci terénu z naměřených dat. Z modelu je možné zjistit výšku každého modelovaného bodu. Je výsledkem procesu, který se nazývá *"surface reconstruction"* nebo *"surface modelling"*[7, kapitola 4.1.1, s. 65].

Matematický model může být vyjádřen ve spojitě nebo prostorově doméně. V prostorově doméně je nejvíce používána funkce dvou proměnných $Z = f(x, y)$. Nejčastěji se používá obecná polynomiální funkce. V praxi je využíváno jenom několik tvarů této funkce.

3.3.1 Datové modely využívané v digitálních modelech terénu

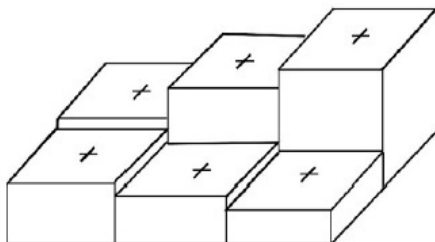
Existuje více způsobů, jak uchovávat data potřebná pro modelování terénů. V dnešní době se nejvíce používají pravidelné a nepravidelné sítě trojúhelníků (TRN - triangular regular network, TIN - triangular irregular network). Dříve byly využívány voxely, což jsou *"samostatné elementy trojrozměrného prostoru"*[5, kapitola 10-16, s. 360]. V následujících podkapitolách budou popsány jednotlivé datové modely.

3.3.1.1 Vrstevnicový model

Jednou z možností, jak reprezentovat data, je pomocí izočar (vrstevnic). Vrstevnice spojují body s konstantní výškou. Jde pravděpodobně o jednu z nejběžnějších metod reprezentace terénu. Přesnost závisí na tom, zda izočary byly generovány z primárního zdroje dat, nebo jsou odvozené. Primární data se vyznačují vysokou přesností, zatímco odvozená jsou interpolována ze známých hodnot sousedních vrstevnic. Hlavní nevýhodou izočar je, že *"zachycují povrch pouze podél izočar"*, a tedy *"nedokáží zachytit anomálie vyskytující se mezi nimi"*[8, kapitola 1.4.1, s. 9]. Protože při digitalizaci izočar z papírových map by bylo nepraktické zachycovat všechny jejich body, digitalizují se pouze *"významné inflexní body"*, takové, při jejichž spojení se *"digitalizovaná vrstevnice významně neodchýlí od skutečné"*[8, kapitola 1.4.1, s. 10].

3.3.1.2 Point-based modelování

Při použití tohoto přístupu je každý bod sítě, který reprezentuje jeden výškový bod, nahrazen vodorovným povrchem určitého geometrického tvaru. Tento povrch zahrnuje nejen samotný bod, ale i jeho okolí. Celý digitální model terénu je tvořen z mnoha vzájemně nespojitých podčástí (viz obr. 3.7). Každá podčást je vyjádřena rovnicí $Z_i = H_i$, kde Z_i je výška i -té podčásti a H_i je výška i -tého bodu v síti, ve které jsou uloženy výšky jednotlivých bodů terénu. Tato metoda je velmi jednoduchá, problém spočívající ve *"vymezení hranic mezi sousedními oblastmi"*[7, kapitola 4.2.2, s. 69] je řešen využitím Voroniových diagramů. V praxi je tento přístup používán, vzhledem k nespojitosti výsledného povrchu, např. v aplikacích, které zobrazují velikost výskytu určité suroviny na určitém území (uhlí, voda).



Obrázek 3.7: Point-based model (převzato z [7, kapitola 4.2.2, s. 69])

3.3.1.3 Pravidelná mřížka

Data jsou ukládána ve formě pravidelné dvourozměrné mřížky (Grid, TIN, triangular regular network). Každý bod mřížky představuje jeden výškový bod na modelu a má čtyři nebo osm sousedů. Body vyskytující se mimo buňky mřížky se dopočítají interpolací okolních bodů. Tvar buňky může podle [7, kapitola 4.2.4, s. 70] být *”čtyřúhelník libovolného tvaru, např. obdélník, čtverec, nebo nepravidelný čtyřúhelník”*, v praxi se ale nejvíce používá pravidelná čtvercová mřížka. Mezi výhody tohoto přístupu patří implicitní přítomnost topologie, snadná implementace a jednoduché zpracování dat.

V případě interpretace členitých terénů je nutné použít velké rozměry mřížky, což může vést k redundanci dat v oblastech, kde terén není variabilní. S tím je spojena i prostorová náročnost uložených dat. Tyto nevýhody lze omezit uložením dat s využitím struktury čtyřstromu (viz kapitola 3.5.1.3) nebo *”run-length”* kódování.

Kompresa *”run-length encoding”* spočívá v nahrazení hodnot sousedících polí se stejnou informací (např. výškou) dvojicí údajů: hodnota pole a souvislý počet polí s touto hodnotou. Kompresa je účinná, pokud se v modelu vyskytují sekvence dat se stejnou hodnotou. Naopak, pokud model představuje členitý terén, jehož výška se často mění, nemusí dojít k žádné úspoře místa. Např. pokud se v modelu vyskytuje následující fragment terénu:

```

1 1 1 1 5 5 3 3 3 3 3
2 2 2 4 4 4 4 5 1 1 1

```


může být s využitím techniky *"run-length encoding"* nahrazen za:

$$\begin{array}{cccc} 1 : 3 & 5 : 2 & 3 : 5 & \\ 2 : 3 & 4 : 4 & 5 : 1 & 1 : 3 \end{array}$$

Mezi výhody datového modelu TIN patří snadné uchovávání dat a manipulace s nimi. Za nevýhody lze považovat riziko vynechání důležitých referenčních bodů v případě příliš velké hrany mřížky, nemožnost přizpůsobit strukturu mřížky dle členitosti terénu a s tím spojená paměťová náročnost.

3.3.1.4 Nepravidelná trojúhelníková síť

Tento model, známý též jako TIN (triangular irregular network), se snaží eliminovat redundanci dat, jednu z nevýhod předchozího modelu. Model není tvořen daty uloženými do pravidelné sítě, ale *"nepravidelně rozmístěnými body, které se mohou přizpůsobit povaze terénu, více bodů je v oblastech s hrubým terénem a méně v oblastech s hladkým terénem"*[8, kapitola 1.4.3, s. 12]. Uložením pouze těch bodů, které jsou z hlediska charakteru terénu důležité, dochází k úspoře místa potřebného pro uložení dat. Naměřené body jsou spojeny do trojúhelníků, což zajistí správnou návaznost sousedících částí a vytvoří spojitý povrch. Oproti předchozímu modelu není implicitně přítomna topologie, a je nutné ji vytvořit ručně. Síť TIN je tvořena třemi elementy: body, úsečkami a trojúhelníky. TIN síť jsou vhodné pro efektivní vyjádření terénů, které lze *"dobře rozdělit na rovné trojúhelníkové plochy"*[8, kapitola 1.4.3, s. 12]. Naopak se nehodí pro modelování povrchů, které není vhodné modelovat plochými trojúhelníky např. u zaledněných krajin.

Za výhody modelu lze považovat možnost popisovat povrch s různými stupni podrobnosti dle charakteru terénu, a nižší paměťové nároky. Nevýhodou může být nutnost provádět ruční výběr dat použitých pro tvorbu sítě.

3.3.1.5 Hybridní přístup

Hybridní přístup představuje kombinaci modelů TRN a TIN. Za základní model je použita pravidelná mřížka. V oblastech, kde je terén členitější, je možné pro jeho modelování použít nepravidelné trojúhelníkové síť.

3.4 Interpolace dat

Datové modely digitálního terénu uchovávají pouze omezený počet bodů se známou informací o jejich poloze (tzv. referenční body). Interpolace je v problematice modelování terénu chápána jako *”zjištění výšky bodu za použití známých hodnot okolních bodů”*[7, kapitola 6.1, s. 115]. Podle [8, kapitola 4.3.1, s. 95] lze interpolaci použít, pokud:

- vymodelovaný terén má jiné rozlišení než je požadováno (je požadováno zhuštění nebo prořidnutí dat),
- je třeba provést změnu formátu dat, např. z TIN do TRN,
- je třeba převést bodová data do formátu, který je možno využít pro analýzu a modelování.

Jednotlivé techniky používané při procesu interpolace lze rozdělit například podle těchto kritérií:

- velikost oblasti interpolace – jsou rozlišovány dvě kategorie: *”area based”* a *”point based”*. V případě přístupu *”area based”* jsou při odhadu povrchu použity *”všechny známé body v určité oblasti”*[7, kapitola 6.1, s. 115]. Kategorii *”area based”* lze dále dělit na globální, tj. konstrukci *”komplexního 3D modelu terénu sestaveného s využitím celé množiny naměřených dat, ze kterého jsou získávány hodnoty všech ostatních bodů”*[7, kapitola 6.1, s. 115], a lokální, kdy *”je větší oblast rozdělena na menší části stejného tvaru a velikosti”* a data jsou interpolována pouze z dat té části, do které spadají[7, kapitola 6.1, s. 115]. Jednotlivé lokální oblasti se zčásti překrývají, čímž je zajištěna spojitost mezi nimi. V přístupu *”point based”* jsou pro interpolaci bodu použity údaje ze všech dostupných referenčních bodů.
- přesnost – pokud jsou při konstrukci povrchu použity hodnoty z referenčních bodů, jde o přesnou rekonstrukci nazývanou *”exact fitting”*. Pokud se výška povrchu liší od hodnoty v referenčních bodech a odchylka od těchto bodů je minimální, jedná se o rekonstrukci *”best fitting”*[7, kapitola 6.1, s. 116] nebo *”inexact interpolation”*[8, kapitola 4.3.1, s. 96].

3.4.1 Exact fitting

3.4.1.1 Lineární interpolace

Pro pravidelné trojúhelníkové sítě lze výšku bodu získat z rovnice

$$z = a_0 + a_1x + a_2y$$

Parametry rovnice lze dopočítat ze soustavy rovnic:

$$a_0 = (1 + x_1 + y_1)^{-1} \cdot z_1$$

$$a_1 = (1 + x_2 + y_2)^{-1} \cdot z_2$$

$$a_2 = (1 + x_3 + y_3)^{-1} \cdot z_3$$

kde $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, $P_3(x_3, y_3, z_3)$ tvoří body roviny trojúhelníku sítě TIN. Pro sítě TRN lze použít postup uvedený v [8, kapitola 4.4.2, s. 119]. Princip spočívá v rozdělení buňky sítě na dva trojúhelníky. Interpolace probíhá následovně:

1. vyhledání buňky, která obsahuje interpolovaný bod (určený dvojicí souřadnic x, y). Tj. určení souřadnic vrcholů buňky s interpolovaným bodem $(x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_4, y_4, z_4)$.
2. normalizace souřadnic interpolovaného bodu podle vzorce:

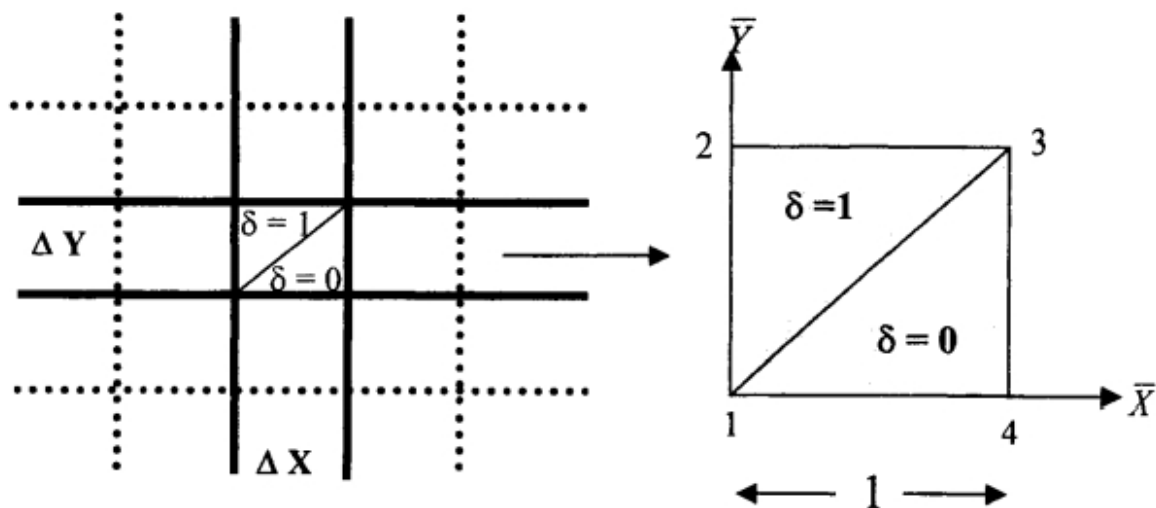
$$X' = \frac{x-x_1}{\Delta x} \quad Y' = \frac{y-y_1}{\Delta y}$$

kde Δx , Δy rovná se šířce, resp. výšce buňky

3. určení ve kterém z trojúhelníků leží interpolovaný bod $\delta = \begin{cases} 1 & \text{pokud } x' < y' \\ 0 & \text{pokud } x' \geq y' \end{cases}$
4. vypočtení výšky interpolovaného bodu dle vzorce:

$$z(x', y') = \delta \cdot (z_1 + (z_3 - z_2) \cdot x' + (z_2 - z_1) \cdot y') + \\ + (1 - \delta) \cdot (z_1 + (z_4 - z_1) \cdot x' + (z_3 - z_4) \cdot y')$$

Uvedený algoritmus generuje spojitý povrch, který ale není hladký.



Obrázek 3.8: Lineární interpolace (převzato z [8, kapitola 4.4.2, s. 120])

3.4.1.2 Bilineární interpolace

Parametry jsou vypočteny z rovnic jednotlivých bodů mřížky $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, $P_3(x_3, y_3, z_3)$, $P_4(x_4, y_4, z_4)$. Vzorce byly převzaty z [8]:

$$\begin{aligned}
 z &= a_0 + a_1x + a_2y + a_3xy \\
 a_0 &= (1 + x_1 + y_1 + x_1y_1)^{-1} \cdot z_1 \\
 a_1 &= (1 + x_2 + y_2 + x_2y_2)^{-1} \cdot z_2 \\
 a_2 &= (1 + x_3 + y_3 + x_3y_3)^{-1} \cdot z_3 \\
 a_3 &= (1 + x_4 + y_4 + x_4y_4)^{-1} \cdot z_4
 \end{aligned}$$

Pokud jsou referenční body uspořádány do tvaru čtvercové mřížky lze pro výpočet použít vzorec:

$$\begin{aligned}
 z_p &= z_1 \left(1 - \frac{\Delta x}{d}\right) \left(1 - \frac{\Delta y}{d}\right) + z_2 \left(1 - \frac{\Delta y}{d}\right) \left(\frac{\Delta x}{d}\right) + \\
 &\quad + z_3 \left(\frac{\Delta x}{d}\right) \left(\frac{\Delta y}{d}\right) + z_4 \left(1 - \frac{\Delta x}{d}\right) \left(\frac{\Delta y}{d}\right)
 \end{aligned}$$

z_p – zjišťovaná výška bodu

z_i – výška i -tého bodu buňky ve které se bod z_p nachází

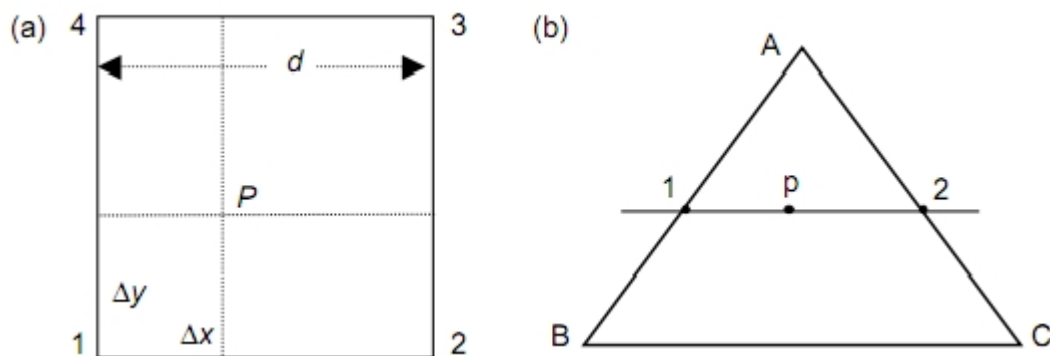
Pro buňky ve tvaru trojúhelníku lze použít podobný postup (viz obr. 3.9):

$$z_p = z_1 + (z_2 - z_1) \cdot (x_p - x_1) / (x_2 - x_1)$$

$$z_1 = z_A + (z_B - z_A) \cdot (x_1 - x_A) / (x_B - x_A)$$

$$z_2 = z_A + (z_C - z_A) \cdot (x_2 - x_A) / (x_C - x_A)$$

kde $y_p = y_1 = y_2$ a body 1 (x_1, y_1, z_1) a 2 (x_2, y_2, z_2) leží na přímkách AB a AC. Pokud se referenční body nacházejí téměř v jedné přímce je vhodnější použít rovnici pro z_p . Výsledkem je spojitý ale nehladký povrch.



Obrázek 3.9: Bilineární interpolace (převzato z [7, kapitola 4.5.1, s. 82])

3.4.1.3 Metoda nejbližšího souseda

Interpolační algoritmus nejbližšího souseda *”přiřazuje interpolovanému bodu výšku náležící nejbližšímu referenčnímu bodu”*[8, kapitola 4.4.2.2, s. 118]. Mezi hlavní výhody tohoto přístupu patří výpočetní nenáročnost, nevýhodou je nespojitost výsledného povrchu.

3.4.2 Best fitting

Tento přístup se používá pokud je *”povrch terénu komplikovaný a je nemožné jej přesně popsat nějakou matematickou funkcí”*[7, kapitola 6.4, s. 123]. Místo toho je použita funkce, která se přesnému popisu tvaru terénu pouze přibližuje. To nemusí nutně znamenat horší model. V případě, že modelovaná oblast je *”rozsáhlá, s mnoha*

dostupnými referenčními body” [7, kapitola 6.4, s. 123] je pro její přesný popis potřeba ”polynomiálních rovnic vyšších stupňů pro dosažení exact-fitting”, což může způsobovat ”náhodné odchylky” [7, kapitola 6.4, s. 123, 6.4]. Přístup ”best-fitting” je v podstatě založen na myšlence, že: ”malé odchylky na povrchu terénu jsou tak složité, že je lze považovat za stochastický proces” [7, kapitola 6.4, s. 123]. Při modelování ”best fitting” je snaha tyto odchylky minimalizovat. Toho lze dosáhnout např. použitím metody nejmenších čtverců:

$$\sum_{i=1}^n e_i^2 = \min$$

kde e_i je odchylka i -tého referenčního bodu od skutečného terénu a n je celkový počet referenčních bodů. Pro jednu množinu referenčních bodů může existovat nekonečné množství řešení best-fitting. V tomto případě se zvolí takové řešení, které má nejmenší součet druhých mocnin e_i .

3.4.3 Globální interpolační metody

Při globální interpolaci jsou k určení výšky bodů mimo oblast známých bodů použity všechny referenční body.

3.4.3.1 Analýza trendu povrchu

Metoda analýzy trendu povrchu při výpočtu výšky požadovaného bodu ”aproximuje povrch polynomiálními rovnicemi geografických souřadnic” a výška ve zkoumaném bodě je považována za ”součet polynomiálních funkcí x a y souřadnic” [8, kapitola 4.4.1.1, s. 97] referenčních bodů (tato složka se nazývá trend) a odchylky (rezidua) matematického modelu od skutečnosti vzniklé při aproximaci zvolenou funkcí. Trend popisuje celkový charakter povrchu, zatímco rezidua lze považovat za lokální detaily. Matematicky lze terén vyjádřit funkcí [8, Kapitola 4.4.1.1, s. 98]:

$$z(x, y) = a_0 + a_1 \cdot X + a_2 \cdot Y + \dots + e$$

$z(x, y)$ – výška zkoumaného bodu

a_i – polynomiální koeficienty

e - rezidua

Koeficienty v rovnicích jsou vypočteny pomocí metody nejmenších čtverců. Za opti-

mální trend je považován takový trend, při kterém je součet reziduí minimální. Obecná rovnice trendu je pak:

$$z(x, y) = \sum_{i=0}^n \sum_{j=0}^{n-1} a_{ij} \cdot x^i \cdot y^j$$

n – řád polynomiální rovnice

Mezi přednosti použití analýzy trendu povrchu patří schopnost *”generovat stejný terén bez ohledu na vybrané referenční body”*, jednoduchá implementace a v případě použití nízkého stupně polynomiální funkce i nenáročnost na výpočetní výkon [8, kapitola 4.4.1.1, s. 107]. K nevýhodám patří předpoklad, že *”aproximovaný povrch je modelován deterministickou funkcí”*, kterého se v praxi podaří dosáhnout jen vyjíměčně. Dále existuje riziko, že *”povrch je příliš komplikovaný na to, aby mohl být vyjádřen polynomiální funkcí nízkého stupně”*[8, kapitola 4.4.1.1, s. 107], na druhou stranu je těžké posoudit *”vhodnost použití konkrétní polynomiální funkce vyššího stupně pro popis trendu”*[8, kapitola 4.4.1.1, s. 107]. Odhad trendu je také závislý na *”vzdálenosti a rozmístění referenčních bodů”*[8, kapitola 4.4.1.1, s. 107], neboť nerovnoměrné rozmístění referenčních bodů může ovlivnit výsledný tvar terénu.

3.5 Zobrazení

Protože zobrazení celé scény je výpočetně náročné, používají se postupy, jak proces vykreslení co nejvíce urychlit. Vykreslování je možné urychlit uložením objektů ve scéně do struktury, ze které lze snadno zjistit, jaké objekty se nacházejí mimo záběr kamery, a tyto objekty odstranit z procesu vykreslování. Tyto struktury se nazývají *”graf scény”*. Jiný způsob urychlení vykreslování, nazývaný *”úroveň podrobnosti”* nebo zkratkou LOD (*level of detail*), spočívá ve snížení detailů zobrazovaného objektu v závislosti na jeho vzdálenosti od pozorovatele. Dalšího zrychlení procesu lze dosáhnout využitím funkcí poskytovaných použitým grafickým API (viz kapitola 3.8).

Kapitola se dále zabývá technikami, které zlepšují vizuální kvalitu zobrazovaného modelu.

3.5.1 Snížení náročnosti renderování scény

Protože vykreslovat všechny objekty nacházející se ve scéně by bylo náročné, a, protože ne všechny objekty se budou nacházet v záběru kamery, také neefektivní, jsou

používány algoritmy, které dokáží množství vykreslovaných objektů zúžit pouze na ty viditelné. Scénu lze považovat za *”množinu prostorových objektů doplněnou dalšími informacemi potřebnými pro jejich zobrazení”*[1, kapitola 14, s. 397]. Tyto informace umožňují *”transformace objektů do jejich cílové podoby”*, jsou *”potřebné pro zobrazování (světla, kamery)”* a dovolují ve scéně používat *”stejně nebo podobně vypadající objekty, tzv. instance”*[1, kapitola 14, s. 397]. Mezi objekty mohou být řazeny:

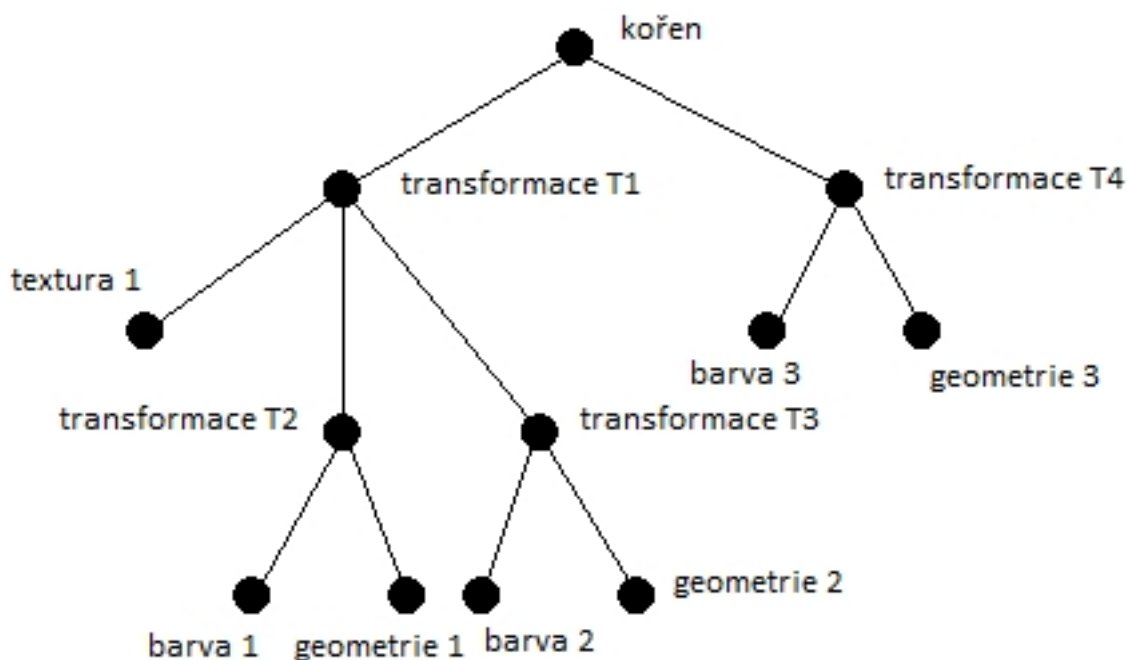
- nezobrazované objekty - kamery, osvětlení,
- zobrazované objekty - jejich geometrie, barevné vlastnosti, textury,
- prvky definující logickou strukturu scény - definice skupin a jejich instancí,
- transformace - definované hierarchicky kvůli snadnější manipulaci s daty.

Je žádoucí, aby objekty byly uspořádány do takové struktury, která dovoluje *”seskupovat logicky k sobě patřící části, efektivně je transformovat a jejich instance vkládat úsporným způsobem do prostoru scény”*[1, kapitola 14, s. 398]. Tato struktura se nazývá graf scény.

3.5.1.1 Graf scény

Graf scény je definován jako n -ární strom, což je strom *”v němž lze pro každý uzel nalézt právě jednoho předchůdce”* [1, kapitola 14.1, s. 398]. Vyjímkou z tohoto pravidla je uzel stromu, který se nachází na nejvyšší úrovni a nemá žádného předchůdce. Uspořádání do stromu umožňuje utvářet vztahy mezi objekty. Je možné například definovat vlastnost a určit, že bude platná pro všechny následovníky uzlu, ve kterém je definována. Touto vlastností mohou být například geometrické transformace (posunutí, otožení, změna velikosti), pomocí kterých je těleso umístěno ve scéně. V jiných systémech, lze rozšířit dědění *”vlastnosti i na uzly na stejné úrovni, umístěné v grafu scény vpravo od daného uzlu”*[1, kapitola 14.1, s. 398]. V těchto systémech je nutné určit pevné pořadí následníků uzlu a stanovit systematické procházení grafu např. zleva doprava.

Na obrázku 3.10 je zobrazen graf scény s třemi objekty (geometrie 1, geometrie 2, geometrie 3), třemi barvami (barva 1, barva 2, barva 3) a jednou texturou (textura 1). Pokud jsou na graf scény aplikovány postupy popsané výše, budou objekty 1 a 2 vykresleny s texturou 1. Každý bude mít jinou barvu, na objekt 1 bude aplikována barva 1 a na objekt 2 barva 2. Objekt 3 bude vykreslen s barvou 3 a nebude na něj aplikována



Obrázek 3.10: Dědění vlastností v grafu scény

žádná textura. Geometrické transformace budou platné pro všechny následníky uzlu, ve kterém je transformace definována, což znamená, že na objekt 1 budou aplikovány transformace T1 a T2, na objekt 2 transformace T1 a T3 a na objekt 3 transformace T4. Při tomto postupu dědění transformací si objekty nemusí ukládat žádné informace o svojí poloze, neboť tu lze získat postupným procházením uzlů s transformacemi předcházejícími uzlu s geometrií tělesa.

S využitím dědičnosti lze ovlivnit působnost světelných zdrojů ve scéně a tím značně urychlit vykreslování. Pokud je známo, že některý světelný zdroj ovlivní pouze určitou část scény, bude umístěn do podstromu reprezentujícího tuto část scény. Lze rozlišovat více druhů stromů dělených podle počtu částí, na kterou scénu rozdělují: BSP (binary space partitioning) strom, čtyřstrom (quad tree), osmistrom (octal tree, octree).

3.5.1.2 BSP stromy

Struktura známá jako BSP strom ukládá scénu do binárního stromu. Strom je vytvářen postupným přidáváním objektů do scény. Uzly představují jednotlivé objekty. Leví následníci každého uzlu představují objekty nacházející se nalevo od objektu

uloženého ve zkoumaném uzlu. Praví následovníci naopak představují objekty nacházející se ve scéně napravo od objektu ve zkoumaném uzlu. Lze tedy říci, že každý uzel rozděluje scénu na dva podprostory. Problém může nastat, pokud dojde ke křížení dvou a více objektů (např. zdí). V takovém případě je nutné objekty rozdělit na více částí tak, aby ke křížení nedocházelo, a každá část náležela do jiného podprostoru scény.

Stromů BSP se využívá ke zjednodušení vykreslování scény nebo při výpočtu kolizí. Mezi omezení stromu patří požadavek, aby objekty, podle kterých je scéna dělena, byly všechny rovnoběžné podél jedné z os. Použití je dále omezeno na statické objekty, neboť přepočítávání stromu je časově náročné.

3.5.1.3 Čtyřstromy a osmstromy

Princip je obdobný jako u BSP stromu, v tomto případě je scéna dělena na čtyři stejně velké oblasti (většinou čtverce), jak lze vidět na obr. 3.11. Každý vrchol stromu reprezentuje informace o příslušné části scény. Pokud data v některé z částí nejsou homogenní, je tato část opět rozdělena na čtyři stejně velké části. Z toho plyne, že každý vrchol má žádného nebo čtyři následovníky. Podle [11] lze dělení zastavit ve třech případech:

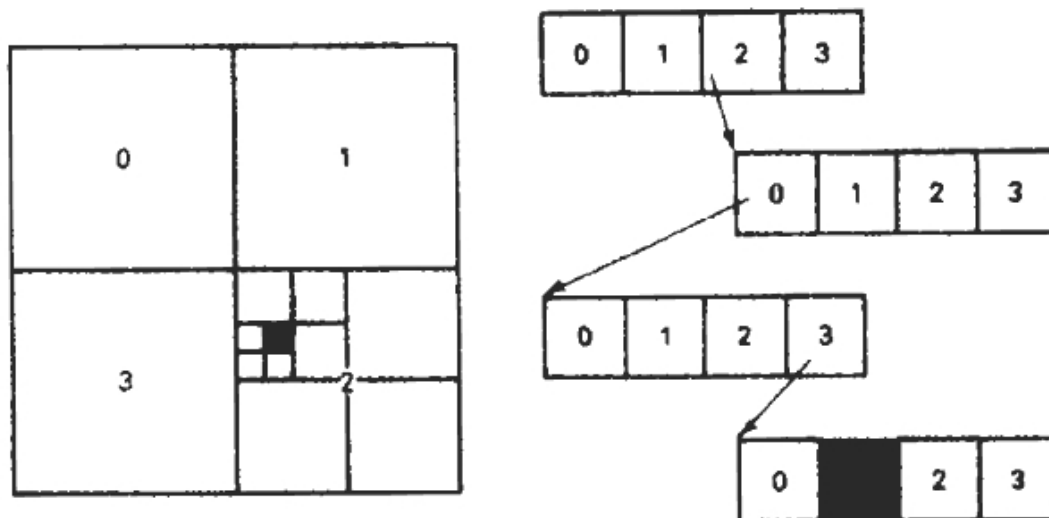
- část obsahuje homogenní data,
- je dosažena určitá hloubka dělení,
- je dosažen určitý počet uzlů stromu.

Místo čtyřstromů je možné využít také osmstromy, které scénu rozdělují na osm kvadrantů. Na rozdíl od čtyřstromů, může každý vrchol mít 0 až 8 následovníků. Pokud některá oblast neobsahuje žádná data k uložení, nebude na ní nastaven odkaz. Tato reprezentace scény může být využita např. pro uložení modelu terénu.

Uvedené struktury se mimo zjednodušení scény využívají i pro testy viditelnosti částí scény nebo detekci kolizí mezi objekty.

3.5.1.4 Úroveň podrobností

Objekty ve scéně se mohou skládat z velkého počtu trojúhelníků, podle množství podrobností, se kterými jsou vymodelovány. Se vzdalováním kamery od objektu se vlivem perspektivy, stejně jako v reálném světě, snižuje množství detailů rozpoznatelných



Obrázek 3.11: Čtyřstrom (převzato z [5, kapitola 10.16, s. 361])

na objektu. Proto je možné každému objektu přiřadit úroveň podrobnosti (LOD - level of detail), s jakou bude zobrazován. Objekt s nejvyšší hodnotou LOD je zobrazen se všemi detaily, jeho geometrie nebude nijak redukována. Naopak čím je hodnota LOD nižší, tím je redukována geometrie objektu a tím více je snižován počet trojúhelníků potřebných k vykreslení. Kritériem pro výběr LOD může mimo vzdálenosti také být *”velikost průmětu objektu na plochu obrazovky, velikost prostorového úhlu, v němž je objekt či jeho obálka vidět z pozice pozorovatele, či celkový počet polygonů v pohledovém objemu”*[1, kapitola 19.2, s. 517].

Při přechodu mezi dvěma úrovněmi podrobnosti může docházet k *”vyskakování částí objektu (popping effect)”* tj. , že *”při zvýšení či snížení stupně detailu se objeví nebo zmizí skupina trojúhelníků”*[1, kapitola 19.2, s. 518]. Tento nežádoucí efekt může být odstraněn např. interpolací geometrií sousedních stupňů podrobnosti.

3.5.2 Skybox

Pojmem skybox se v počítačové grafice rozumí objekt, který nese texturu okolního prostředí, a obklopuje renderovanou scénu, čímž je dosaženo zdání, že scéna není umístěna volně v prostoru, ale je součástí okolí.

Za obklopující objekt bývá nejčastěji zvolena krychle (skybox) nebo koule (sky-

sphere). Pokud je za objekt zvolena krychle, je nutné vybrat vhodné textury tak, aby dobře navazovaly v místech přechodu mezi sousedními stěnami krychle.

Skybox může být do scény umístěn staticky, nebo se může pohybovat (pouze pohybovat, nikoly rotovat) v závislosti na pohybu kamery tak, aby byla umístěna vždy v jeho středu. Na obrázku 3.12 je uveden příklad textury použitelné pro skybox, a výsledná scéna s implementovaným skyboxem.

3.6 Využití v praxi

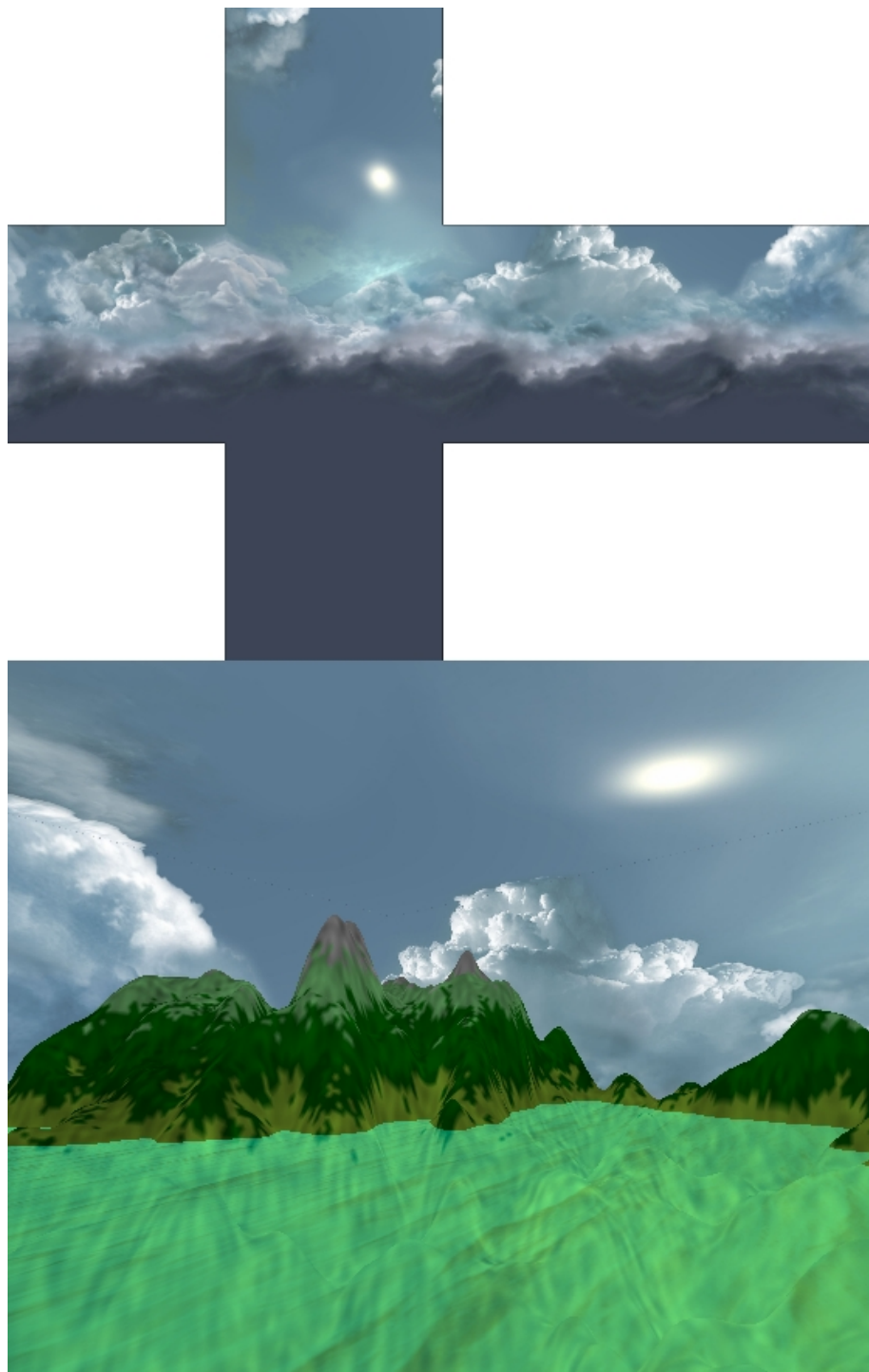
Digitální modely terénu a jejich modelování nachází v současné době mnohá praktická uplatnění. Modely jsou použity ve všech částech ekosystému Země, nalézají uplatnění v modelování *”atmosféry (klimatické proměnné), hydrosféry (pozemní a podzemní toky), a pedosféry (vegetace a eroze)”*[8, kapitola 8, s. 207]. Digitálních modelů terénu je využíváno v mnoha odvětvích.

3.6.1 Stavebnictví

Digitální modely terénu nacházejí mnoho aplikací při stavbě silnic, železnic, kanálů, přehrad atd. Modely se používají pro vybrání vhodné polohy stavby nebo plánování nejvhodnější trasy.

3.6.1.1 Navrhování železnic a silnic

Proces výstavby dopravní komunikace, ať silnice či železnice, lze rozdělit do několika kroků[7]: průzkum terénu, plánování a návrh trasy, výpočet rozsahu úpravy terénu, návrh tunelů a mostů atd. Z těchto kroků se modely terénu uplatňují především při plánování a návrhu trasy a při výpočtu rozsahu úpravy terénu. Modely slouží k zajištění takové trasy, která má *”stabilní geologické podmínky, pozvolné svahy a malé zakřivení”*[7, kapitola 14.1.1, s. 286], což vede k minimalizaci terénních úprav. Dále jsou modely používány pro tvorbu perspektivních pohledů na stavbu nebo navrhování křižovatek. Mezi základní požadavky na modely používané v tomto odvětví patří *”možnost vkládat a odebírat triangulační body a přiřazovat jim určitou výšku”*[7, kapitola 14.1.1, s. 286].



Obrázek 3.12: Skybox: použité textury a výsledná scéna

3.6.1.2 Modelování odtokových systémů

Další oblastí, ve které se využívají digitální modely terénu jsou návrhy kanálů, vodních nádrží a dalších objektů v krajině, které souží pro vedení nebo zadržování vody. Při návrhu nádrží může model pomoci určit kritéria jako [7]: pozice stavby, množství zadržené vody, velikost odtoku apod. Aby mohly být tyto stavby zkonstruovány na správném místě, jsou vytvářeny na základě modelů zkoumajících povodí v dané oblasti. Povodí je definováno jako *”oblast, ve které všechna voda stéka do jednoho místa”*[8, kapitola 8.1, s. 208]. Toto místo je také nazýváno *”odtokem povodí”*. Digitální modely terénu jsou využívány pro definování hranic povodí a modelování hromadění vody v oblasti.

3.6.2 Výzkum planety Země

Obory zabývající se zkoumáním přírodních jevů probíhajících na planetě Zemi využívají modely terénu pro předpovědi počasí, modelování klimatu nebo sledování pohybu nižších atmosférických vrstev. Analýza terénu je využívána také v oblastech hydrologie nebo geologie pro tvorbu srážkových map nebo modelů možných následků přírodních katastrof.

3.6.2.1 Povětrnostní modely

Povětrnostní modely jsou důležité pro předpovědi počasí, předpovědi směru šíření lesních požárů nebo znečištění. Směr a síla větru může být ovlivněna[7]: výškou, sklonem, směrem a hrubostí terénu. Všechny tyto informace je možné získat z digitálních modelů terénu.

3.6.2.2 Modelování rizika vzniku požárů

Často se stává, že pro pochopení modelovaných jevů je terén chápán jako *”nezbytný vstup”*, který *”je nutné doplnit s ostatními zdroji informací”*[8, kapitola 8.9.4, s. 237]. Informace z každého zdroje lze zobrazit do různých vrstev, které překrývají terén. Kombinací údajů v těchto vrstev vznikne výsledný údaj modelující zkoumaný jev.

Jako příklad lze uvést model určující riziko vzniku požárů v závislosti na několika atributech. Mezi atributy lze řadit[8]: převládající směr větru, sklon svahů a hustotu vegetace na těchto svazích. Svahy, které *”jsou skloněny proti větru a jsou porostlé*

hustou vegetací”[8, kapitola 8.9.4, s. 237] mohou být považovány za rizikové oblasti. Z těchto a dalších faktorů může být vypočten koeficient rizika vzniku požáru a zobrazen do modelu.

3.6.3 Plánování a správa surovin

Modely terénu jsou využívány také v oblastech, které se zajímají o přírodní zdroje. Jedná se např. o oblasti vzdáleného průzkumu Země, zemědělství, lesnictví, plánování výstavby měst atd. V zemědělství mohou být modely použity pro modelování eroze půd, šíření znečištění aj.

3.6.4 Vojenství

Armáda je jedním z největších producentů a zároveň spotřebitelů digitálních modelů terénu. Terény jsou využity jako data do leteckých simulátorů pro výcvik pilotů a plánování misí. Armáda dále využívá modely terénu pro navigaci inteligentních střel.

3.7 Použité technologie

Při psaní praktické části práce byly autorem použity následující technologie:

- **OpenGL** (Open Graphics Library) – jedná se o knihovnu funkcí pro práci s grafickým hardwarem. Knihovna je nezávislá na hardware, tato vlastnost je dosažena přítomností pouze základních funkcí pro renderování grafiky. Knihovna neposkytuje žádné funkce pro práci se systémem oken, ani pro vykreslování složitějších objektů. Od verze 2.0 je součástí OpenGL jazyk GLSL (OpenGL Shading Language) pro psaní programů běžících na grafické kartě (shadery). V době psaní této práce je dostupné OpenGL verze 4.2. Autor si OpenGL zvolil, protože s tímto API má několikaleté zkušenosti.
- **.NET framework** – API navržené pro usnadnění tvorby programů pod platformou Windows. .NET framework je postaven na principech objektově orientovaného programování. Programy napsané v jazycích podporovaných platformou .NET jsou nejdříve zkompileovány do MSIL (Microsoft intermediate Language) a teprve ten je přeložen do strojového kódu, čímž je umožněna tvorba aplikace

v různých jazycích a teoreticky i přenositelnou na jiné platformy. Pro usnadnění vývoje aplikací poskytuje .NET Framework širokou škálu tříd.

- **C#** – Objektivě orientovaný programovací jazyk pro vývoj aplikací na platformě .NET se silnou typovou kontrolou. Jazyk byl speciálně navržen pro práci na platformě .NET. Důvodem, proč autor použil tento jazyk, je rozsáhlá knihovna tříd, která usnadňuje programování mnoha běžných operací. Pro C# je dostupné oficiální vývojové prostředí Microsoft Visual Studio, které, mimo jiné, obsahuje nástroje pro snadnou tvorbu uživatelského rozhraní prostřednictvím formulářů a komponent.
- **OpenTK** (Open Toolkit Library) – Knihovna umožňující pracovat s OpenGL v jazyku C#.

3.8 Grafické API OpenGL

OpenGL představuje množinu funkcí pro komunikaci s grafickým hardware. Vnitřně lze OpenGL považovat za stavový automat. Pokud je nastaven nějaký atribut, např. osvětlení, jsou všechny vykreslované objekty tímto atributem ovlivněny, dokud není deaktivován. Pro práci OpenGL využívá několik paměťových bufferů. Data z nich mohou být zobrazena na výstupním zařízení (buffer barev), sloužit k odstranění skrytých povrchů objektů (buffer hloubky) nebo být využívána pro pokročilé renderovací techniky (antialiasing celé scény).

Než je grafické primitivum (fragment) vykresleno na výstupním zařízení, projde postupně sérií operací, které mohou ovlivnit jeho vlastnosti, případně jej úplně odstranit, ještě před uložením do bufferu barev. Např. původní barva fragmentu může být ovlivněna mlhou, osvětlením, operacemi míchání barev (blending) a dalšími funkcemi. Více informací o funkcionalitě OpenGL lze nalézt na oficiálních stránkách OpenGL (www.opengl.org/) nebo v [2].

3.8.1 Míchání barev (blending)

Operace míchání umožňuje určit, zda pixel v bufferu barev (tzv. cílový pixel) bude nově příchozím (zdrojovým) pixelem přepsán, nebo se jejich hodnoty smíchají v závislosti na aktuálním nastavení OpenGL. Jenodtlivé barevné složky výsledného pixelu

KAPITOLA 3. PROBLEMATIKA POČÍTAČOVÉHO MODELOVÁNÍ

jsou vypočteny, pokud není nastaveno jinak, z rovnice[2, kapitola 6, s. 211]:

$$R_s \cdot S_r + R_d \cdot D_r, G_s \cdot S_g + G_d \cdot D_g, B_s \cdot S_b + B_d \cdot D_b, A_s \cdot S_a + A_d \cdot D_a$$

R_s, B_s, G_s, A_s - barevné složky zdrojového pixelu

R_d, B_d, G_d, A_d - barevné složky cílového pixelu

S_r, S_g, S_b, S_a - zdrojový faktor

D_r, D_g, D_b, D_a - cílový faktor

Za zdrojový a cílový faktor lze dosazovat hodnoty předdefinovaných konstant použitím příslušných funkcí OpenGL. Hodnoty konstant jsou dostupné např. v [2]. Pomocí míchání barev lze vytvářet průhledné objekty nebo implementovat techniku maskování.

Kapitola 4

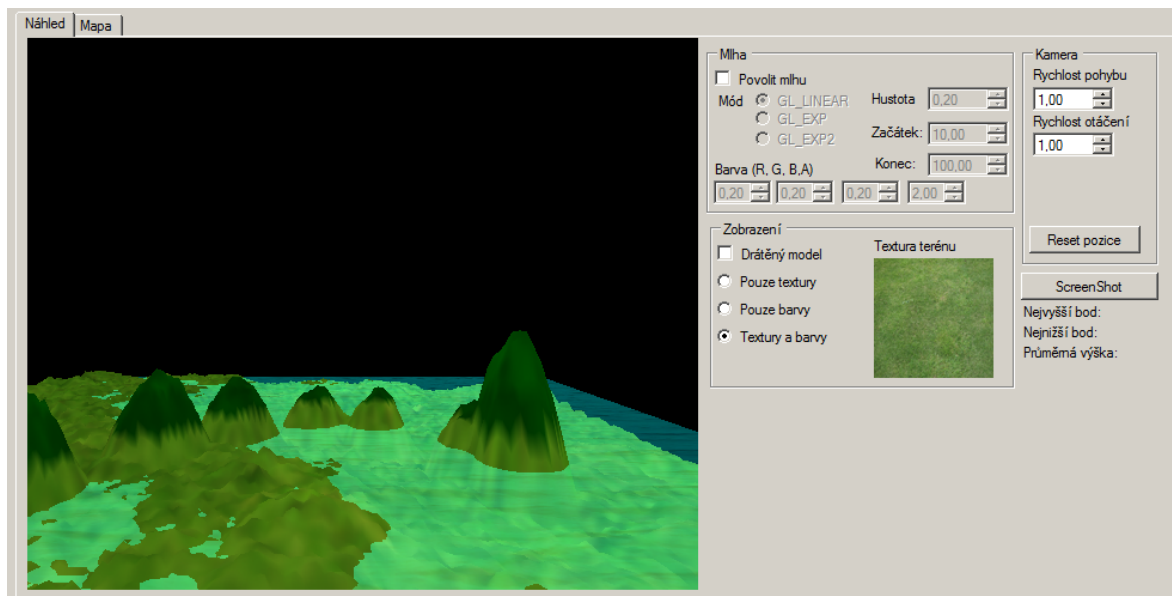
Implementace programu pro generování a zobrazení 3D terénu

Praktická část diplomové práce si klade za cíl navržení a implementaci programu, který předvede některé techniky pro vytváření a úpravy digitálního modelu terénu. Navržený program obsahuje editor terénů, jehož součástí je i náhled vygenerovaného terénu v podobě 3D modelu.

4.1 Popis uživatelského rozhraní

Uživatelské rozhraní programu je rozděleno do dvou částí. V první části (viz obr. 4.1) se nacházejí ovládací prvky pro změnu parametrů renderingu scény a okno s náhledem vygenerovaného terénu. Po scéně se lze pohybovat pomocí kláves w (pohyb dopředu), s (pohyb dozadu), a (otočení doleva), d (otočení doprava), q (pohyb nahoru), z (pohyb dolů). Podržením levého tlačítka a pohybem myši lze měnit úhel pohledu. Funkce pro nastavení renderingu umožňují zapnout mlhu a měnit její parametry (barva, hustota, použitý algoritmus pro generování, počátek a konec mlhy). Napravo od prvků pro nastavení vlastností mlhy je umístěna skupina prvků pro změnu rychlosti pohybu a rotace kamery. Kliknutím na obrázek v části "Zobrazení" lze nastavit texturu aplikovanou na terén. Ve spodní části okna jsou umístěny vybrané informace o výšce bodů terénu.

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU



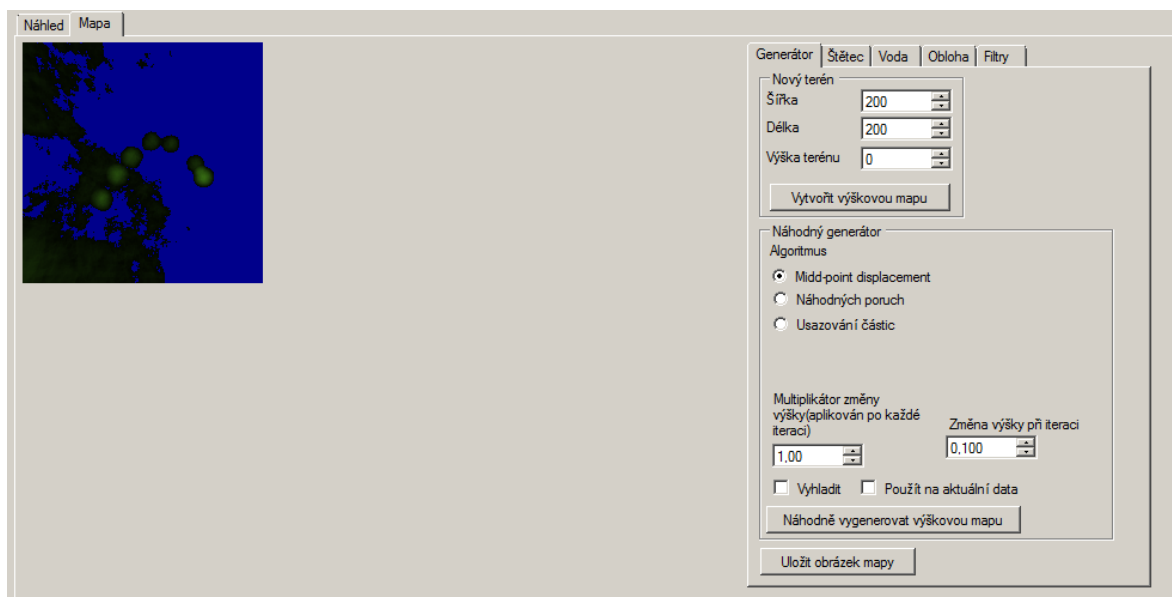
Obrázek 4.1: Nastavení vykreslování

Ve druhé části (viz obr. 4.2) pod záložkou "Mapa" jsou umístěny prvky pro nastavení generátoru terénu, vody, skyboxu a vyhlazování.

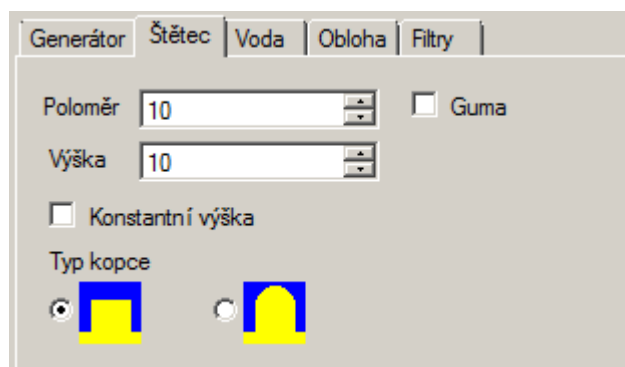
Pod záložkou "Generátor" jsou umístěny prvky pro nastavení parametrů náhodného generátoru umožňující měnit velikost terénu, algoritmus generování a jeho parametry.

Pod záložkou "Štětce" (viz obr. 4.3) se nacházejí prvky pro změnu poloměru, výšky a typu štětce pro tvorbu kopců a údolí. Levým kliknutím na mapu nebo tažením myši se stisknutým levým tlačítkem se v terénu vytváří kopce. Pravé tlačítko slouží k tvorbě údolí.

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU



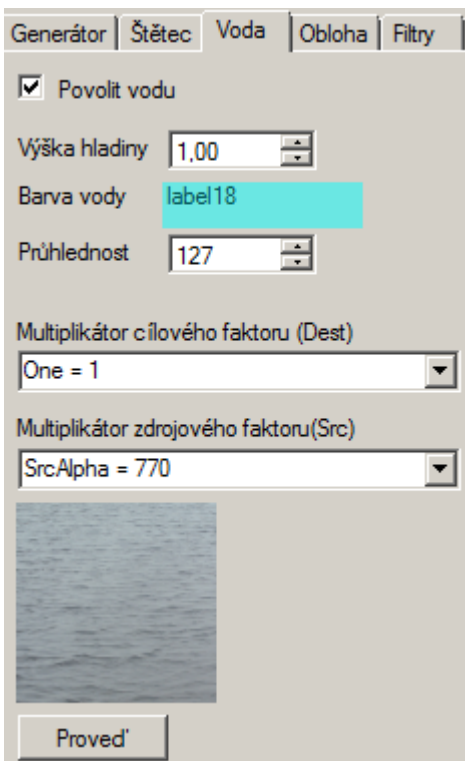
Obrázek 4.2: Nastavení generátoru



Obrázek 4.3: Nastavení štětce

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU

Záložka "Voda" (viz obr. 4.4) umožňuje povolit vykreslování vodní hladiny, nastavit její barvu a průhlednost, texturu, výšku a měnit zdrojový a cílový faktor pro rovnici blendingu.



Obrázek 4.4: Nastavení vodní hladiny

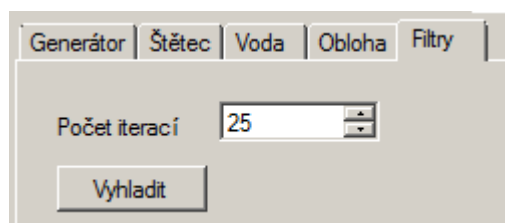
Záložka "Obloha" (viz obr. 4.5) obsahuje ovládací prvky pro vykreslování skyboxu a nastavení textur pro jeho jednotlivé stěny.

Záložka "Vyhlazování" (viz obr. 4.6) obsahuje funkce pro vyhlazování vygenerovaného terénu

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU



Obrázek 4.5: Nastavení skyboxu



Obrázek 4.6: Nastavení vyhlazování

4.2 Editor terénů

Autorem navržený editor je rozdělen na několik částí. První část slouží k náhledu vygenerovaného terénu, v dalších částech se nacházejí prvky pro náhodné generování a úpravy terénu, nastavování parametrů vodních ploch, skyboxu, a vyhlazování.

4.2.1 Náhodný generátor terénu

Pro náhodné generování terénu byly autorem implementovány tři algoritmy: přesouvání středního bodu, usazování částic a metoda náhodných poruch. Jejich popis najdete v teoretické části práce (viz kapitola 3.2.5). Výslednou podobu terénu je možné ovlivnit úpravou parametrů před zahájením procesu generování. Výsledek procesu je zobrazen v podobě výškové mapy.

4.2.1.1 Přesouvání středního bodu

Základním principem metody je rekurzivní opakování dvou kroků: změna výšky bodu uprostřed terénu (*midpoint*) a rozdělení terénu na čtyři stejně velké podoblasti (viz obr. 3.3). Na každou nově vzniklou oblast je pak aplikován stejný postup. Průběh algoritmu lze ovlivnit nastavením:

- velikosti náhodné změny výšky středního bodu (delta),
- multiplifikátorem parametru delta - parametr delta je tímto parametrem vynásoben při každé iteraci, čímž lze dosáhnout postupného zjemňování, nebo naopak větší hrubosti změn v terénu.

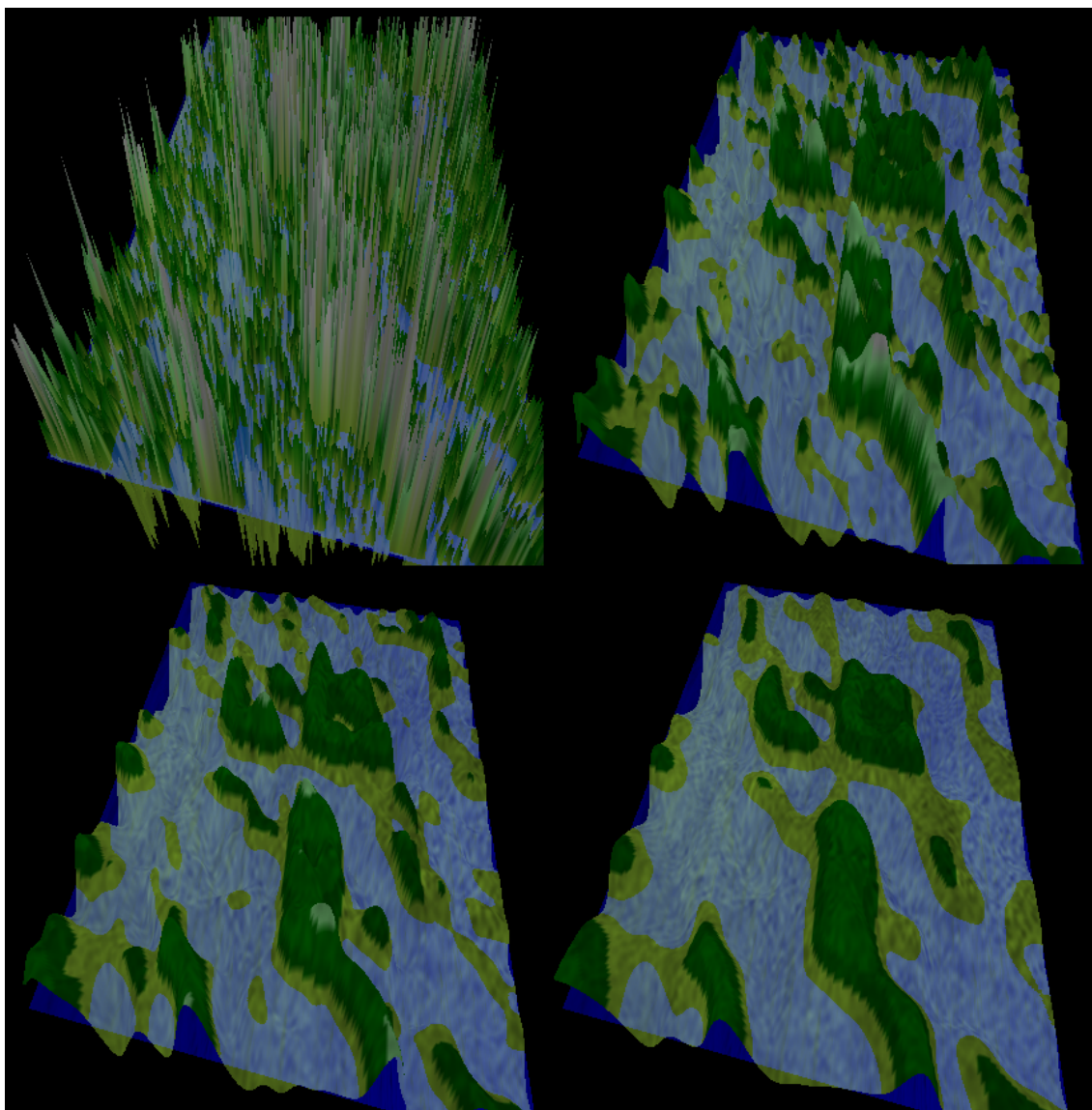
Pro demonstraci výsledků autor vygeneroval dva modely, každý s jinými hodnotami parametrů (viz tabulka 4.1). Každý z modelů byl následně vyhlazen několika iteracemi vyhlazovacího algoritmu (viz kapitola 4.2.2), aby se zmírnila hrubost a dosáhlo se přirozenějšího vzhledu terénu. Počet iterací je u všech modelů shodný: 0, 10, 30, 80.

Model	Delta	Multiplikátor
1	20	1
2	20	0,8

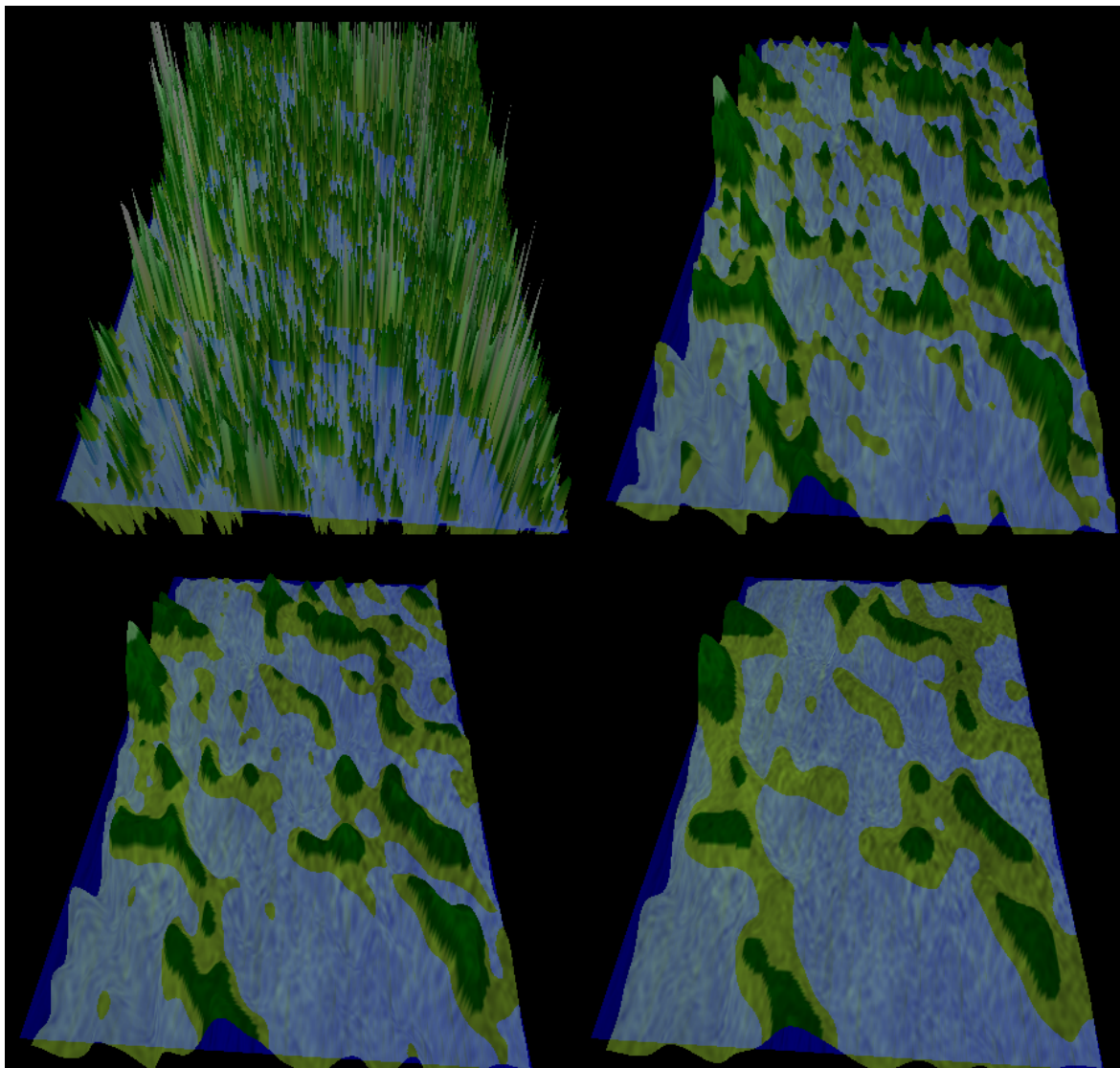
Tabulka 4.1: Metoda přesunu středního bodu - parametry modelů

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU

Z obrázků 4.7 a 4.8 je zřejmé, že generovaný model je bez aplikace vyhlazování nepoužitelný. Prvních deset iterací vyhlazování má nejvyšší vliv na strukturu terénu. Algoritmy generují značně chaotický terén, avšak lze si všimnout, že v něm dochází ke vzniku kratších či delších pohoří.



Obrázek 4.7: Přesun středního bodu - model č. 1, vpravo po vyhlazení



Obrázek 4.8: Přesun středního bodu - model č. 2, vpravo po vyhlazení

4.2.1.2 Metoda náhodných poruch

Mezi základní principy metody patří rozdělení roviny na dvě části nějakým dvou-rozměrným objektem. V programu autor za tento objekt zvolil náhodně umístěnou přímku protínající rovinu terénu. Příмка je určena dvěma náhodně umístěnými body. Výsledek generování lze ovlivnit několika vstupními parametry:

- počtem iterací (počtem přímek, kterými bude rovina terénu postupně protnuta),
- o kolik se zvýší resp. sníží výška terénu po stranách přímky (parametr delta),
- multiplikátorem parametru delta - jeho význam je stejný jako u předchozího algoritmu.

Pro demonstraci výsledků autor vygeneroval čtyři modely, každý s jinými hodnotami parametrů (viz tabulka 4.2). Každý z modelů byl následně vyhlazen několika iteracemi algoritmu pro vyhlazování.

Model	Počet iterací	Delta	Multiplikátor	Iterace vyhlazování
1	1000	0,1	1	0,10
2	1000	1	1	0,20,30,80
3	2000	1	0,2	0,20,30,80
4	10 000	1	0,9	0,60,100,200

Tabulka 4.2: Metoda náhodných poruch - parametry modelů

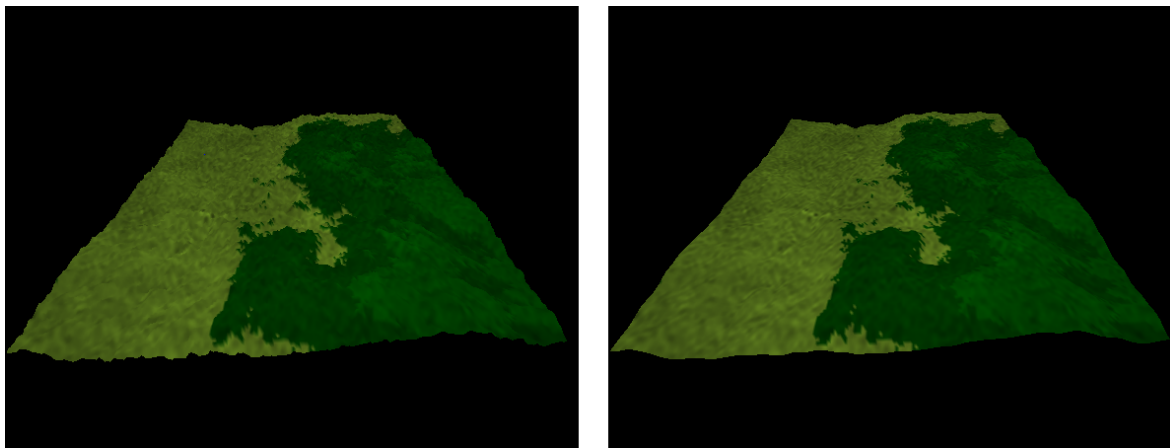
První model (obr. 4.9) – Kombinací poměrně nízkého počtu iterací a parametru delta blízkému nule lze vygenerovat málo členitý terén s nízkým převýšením. Na pravém obrázku je tentýž model po aplikování 10 iterací vyhlazovacího algoritmu. Nízký počet iterací vede pouze k malému vyhlazení (lze si všimnout zjemnění u spodní hrany terénu).

Druhý model (obr. 4.10) – Při tvorbě modelu byl zvýšen parametr delta, což se projevilo ve vzniku výrazně členitějšího terénu s vysokým převýšením. Na model je postupně aplikován algoritmus pro vyhlazování, jehož působení je nejlépe viditelné především na ostrých vrcholcích hor.

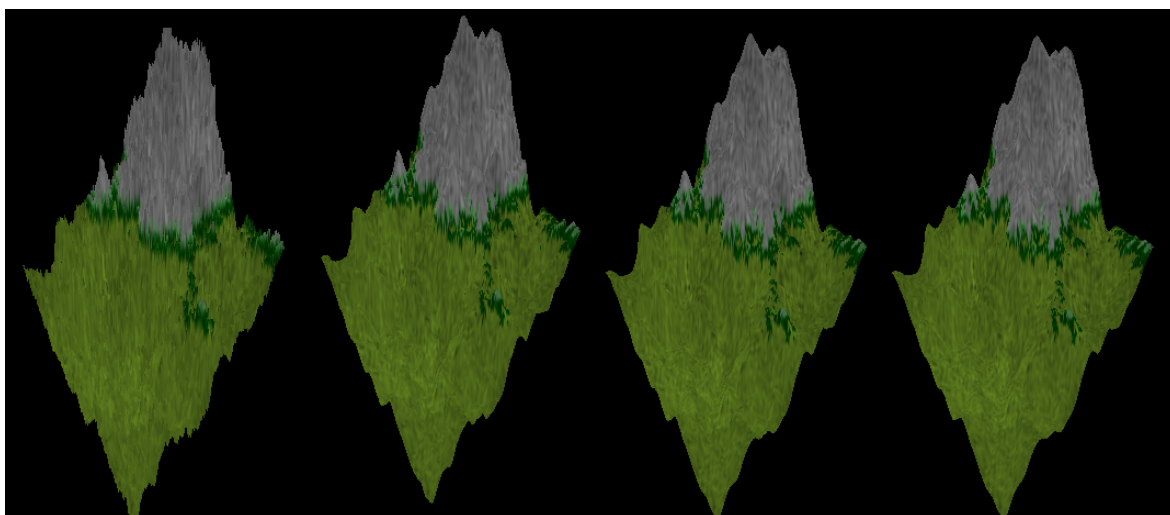
Třetí model (obr. 4.11) – Snížením multiplikátoru parametru delta došlo k vytvoření strmějších kopců a horských vrcholků.

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU

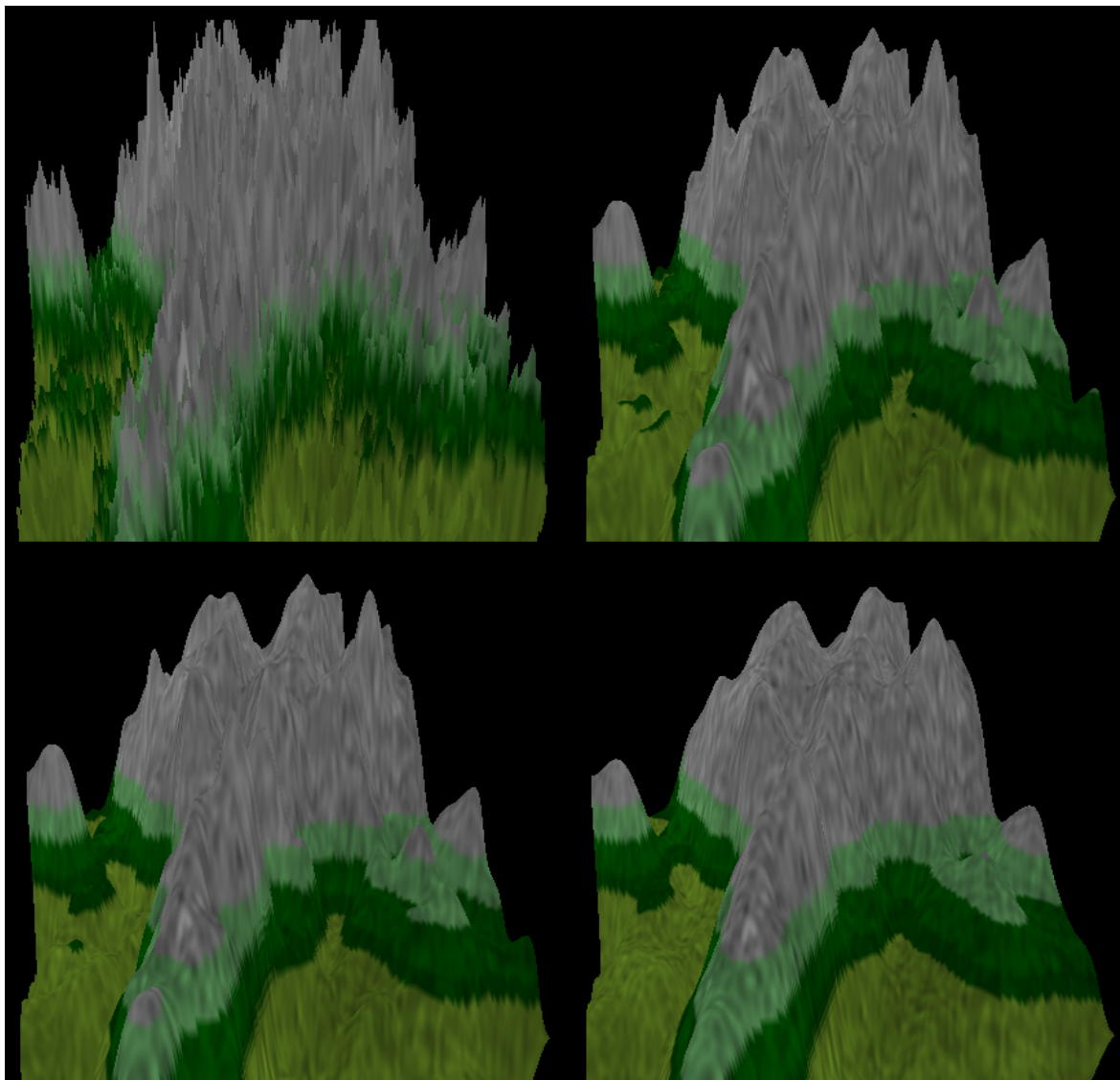
Čtvrtý model (obr. 4.12) – Zvýšení počtu iterací vedlo ke vzniku několika oddělených vrcholků hor. Po aplikaci několika průchodů algoritmem pro vyhlazování si lze všimnout postupného zaoblování ostrých vrcholků hor.



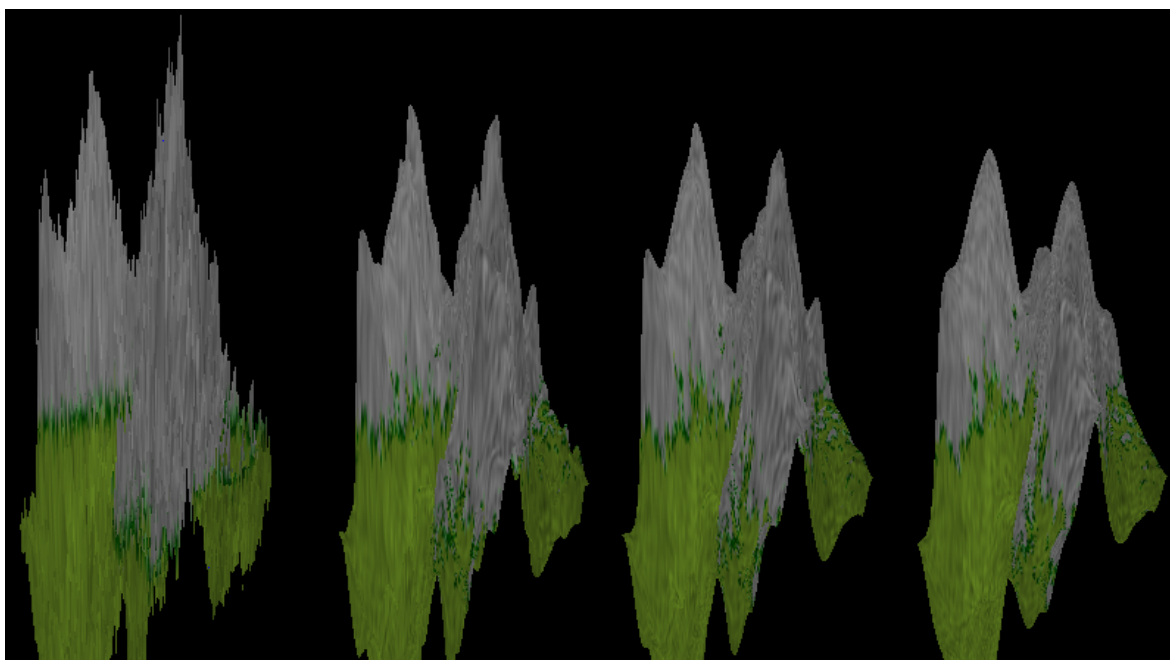
Obrázek 4.9: Metoda náhodných poruch - model č. 1, vpravo po vyhlazení



Obrázek 4.10: Metoda náhodných poruch - model č. 2, vpravo po vyhlazení



Obrázek 4.11: Metoda náhodných poruch - model č. 3, vpravo po vyhlazení



Obrázek 4.12: Metoda náhodných poruch - model č. 4, vpravo po vyhlazení

4.2.1.3 Usazování částic

Dalším implementovaným algoritmem je metoda usazování částic. Její průběh lze ovlivnit:

- počtem iterací (počet částic),
- o kolik jednotek se terén zvýší v místě dopadu částice (energie částice),
- o kolik jednotek se energie částice sníží po každém přesunu na novou pozici (úbytek energie),
- velikostí okolí do kterého se částice může přesunout ze své aktuální pozice.

Částice se může přesouvat pouze do poloh s nižší výškou než je její aktuální výška. Pokud se částice při svém pohybu dostane mimo oblast terénu, nebo se v jejím okolí nenachází žádný soused, na kterého se může přesunout, je smazána a dále již terén neovlivňuje.

Pro demonstraci výsledků autor vygeneroval 2 modely, každý s jinými hodnotami parametrů (viz tabulka 4.3). Pro dobré výsledky je vhodné zadávat počet iterací nejméně 1000. Každý z modelů byl následně upraven vyhlazovacím algoritmem.

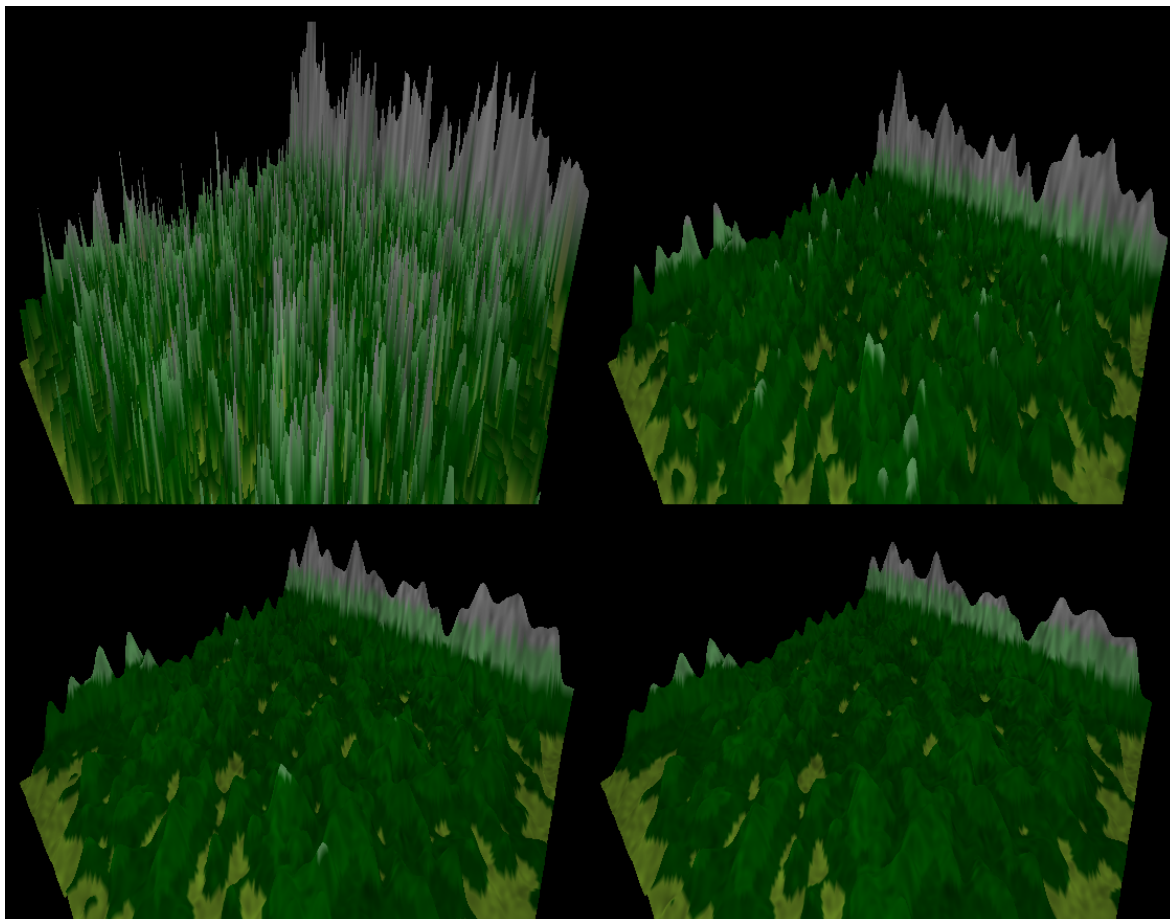
Pro dosažení hladkého terénu stačí relativně malý počet iterací v porovnání s počtem iterací nutným pro vyhlazení terénu generovaného metodou náhodných poruch. Nastavení hodnoty na 20 iterací poskytuje dostatečně hladký terén.

Model	Počet iterací	Velikost okolí	Energie částice	Úbytek energie
1	1000	1	10	0,001
2	1000	1	20	1

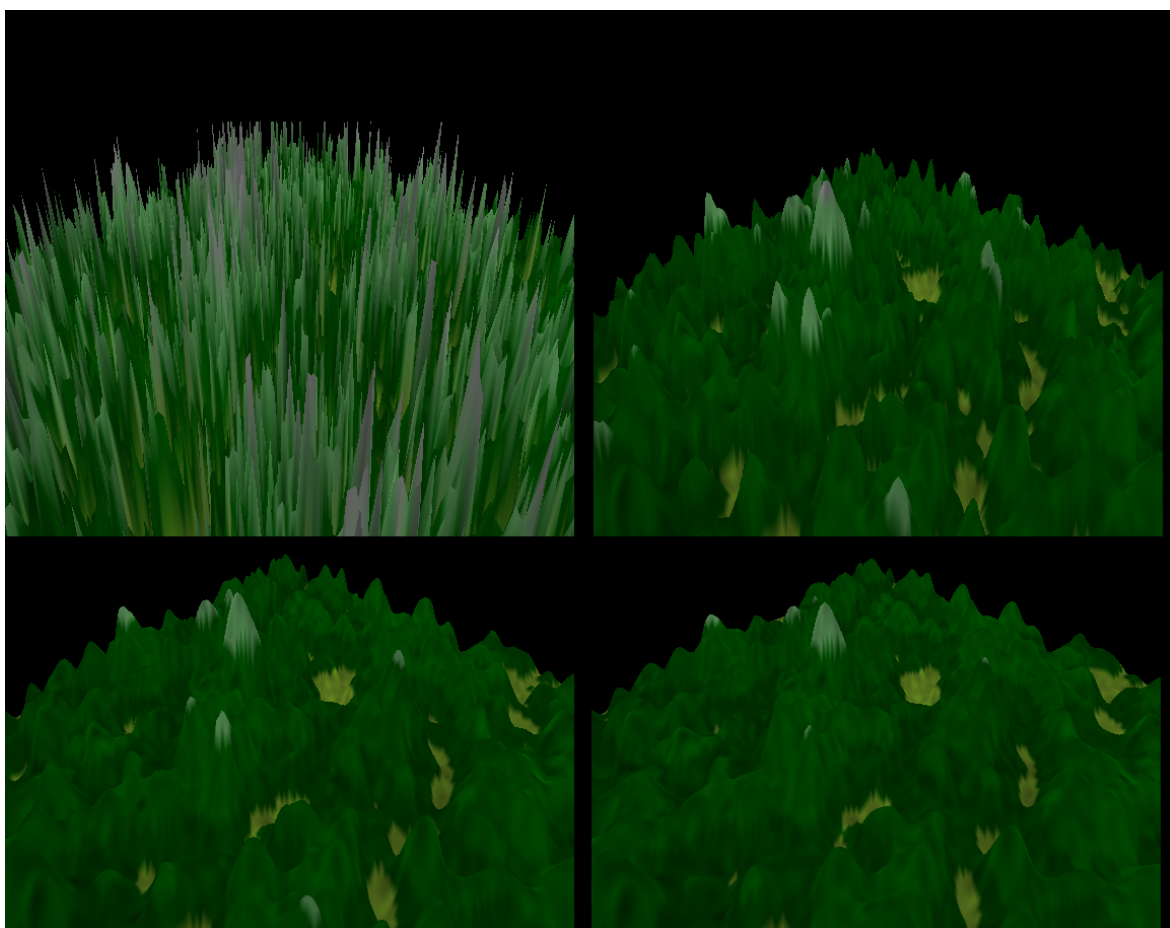
Tabulka 4.3: Metoda usazování částic - parametry modelů

Z obrázků 4.13 a 4.14 je vidět, že metoda usazování částic generuje poměrně členitý a kopcovitý terén. Ve druhém modelu byl zvolen rychlejší úbytek energie částice, což se projevilo vznikem vyšších a strmějších vrcholků v terénu.

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU



Obrázek 4.13: Metoda usazování částic - model č. 1, vpravo po vyhlazení



Obrázek 4.14: Metoda usazování částic - model č. 2, vpravo po vyhlazení

4.2.2 Vyhlazování terénu

Protože funkce pro náhodné generování terénu vracejí terén s nepřírozeně ostrými vrcholy, byla autorem do programu implementována funkce pro vyhlazování povrchu terénu. Funkce zaoblí ostré hrany a útvary v terénu, takže nedochází k prudké změně směru normálového vektoru sousedících polí (rovin) terénu, ale s rostoucím počtem iterací se snižuje výšková rozmanitost terénu. Optimální počet iterací byl autorem empiricky stanoven na hodnotu okolo 25, avšak záleží na konkrétním modelu terénu a uživateli, po kolika iteracích vyhlazování bude s výsledkem spokojen.

Pokud by se počet iterací vyhlazování blížil nekonečnu, byl by výsledkem plochý terén. Autor tuto teorii empiricky ověřil nastavením počtu iterací na 300 000.

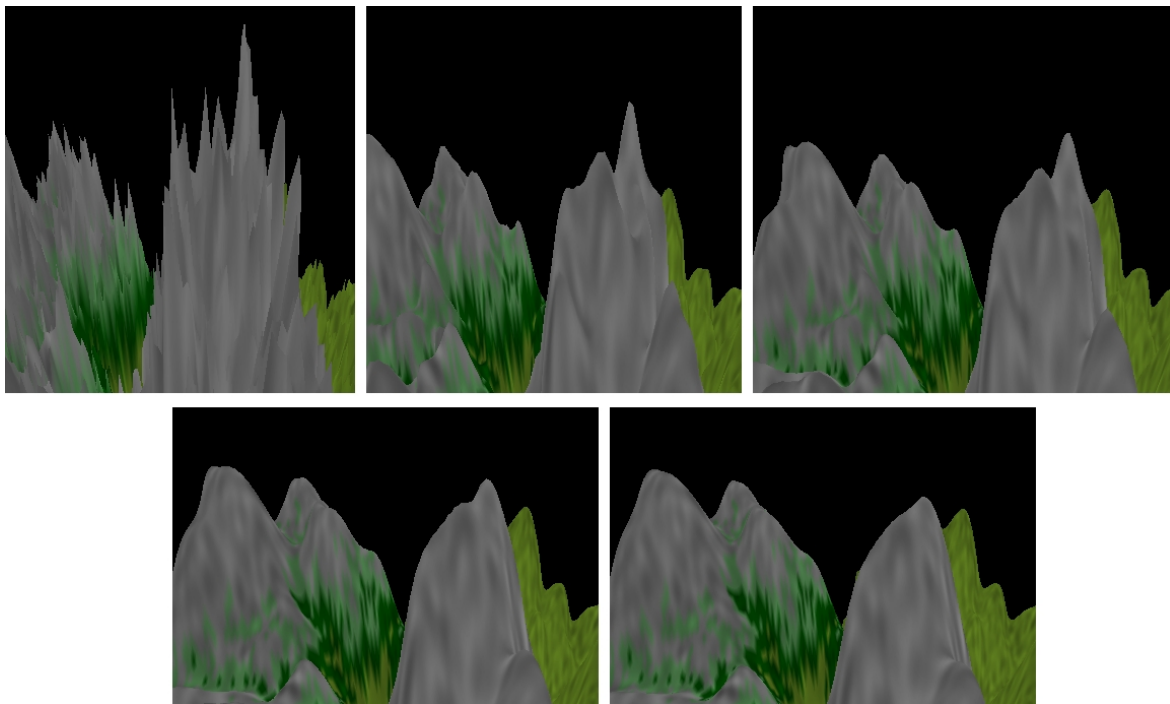
Použitý algoritmus byl převzat z [12]. Algoritmus postupně projde všechny vrcholy terénu a nad každým provede následující kroky:

1. výpočet průměrné výšky (S) sousedních vrcholů aktuálního vrcholu (A),
2. výpočet průměru z S a výšky vrcholu A ,
3. uložení výsledku z kroku 2. do nového pole.

Nově vypočtené hodnoty jsou ukládány do samostatného pole, aby neovlivňovaly výšku vrcholů dosud nezapočtených v aktuální iteraci. Průběžné výsledky algoritmu (vždy po 10 iteracích) jsou znázorněny na obrázku 4.15.

4.2.3 Ruční úpravy terénu

Mimo náhodného generování je do programu autorem implementována možnost ručních úprav tvaru terénu. Tyto úpravy zahrnují tvorbu kopců a údolí. V aplikaci jsou implementovány algoritmy pro tvorbu dvou typů kopců. Údolí jsou vytvářena stejnými algoritmy jako kopce, pouze místo přičítání, se výška v místě umístění odečítá. Všechny typy kopců jsou určeny svým poloměrem a výškou. Výškou je myšlena výška nejvyššího bodu útvaru. Za poloměr je považován poloměr kruhu výškových bodů, které jsou ovlivněny při tvorbě útvaru.



Obrázek 4.15: Proces vyhlazování po 0, 10, 20, 30 a 40 iteracích

4.2.3.1 Metoda MakeHillFlat

Metoda nastaví výšku všech bodů v poloměru útvaru na zadanou hodnotu. Výsledkem je kopec s plochým vrcholem. Výška je tedy dána předpisem:

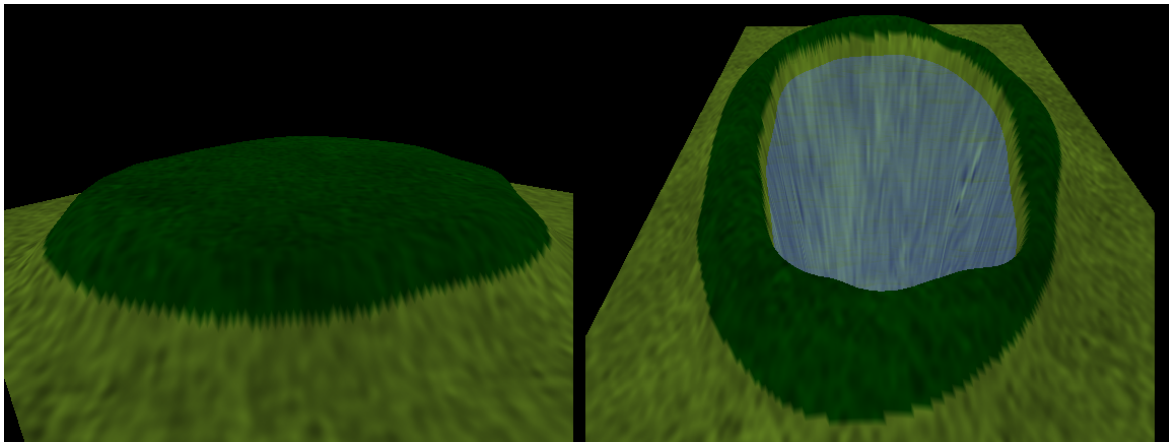
$$y = h$$

h - výška kopce

Pro věrohodnější výsledky je žádoucí vzniklý útvar vyhladit. Na obrázku 4.16 je zobrazen kopec vygenerovaný s poloměrem a výškou rovnou 10 jednotkám. Pravý obrázek ukazuje jak, vytvořením prohlubně v místě vrcholu kopce a nastavením výšky vodní hladiny (viz kapitola 4.2.5) pod úroveň terénu, vytvořit jezero.

4.2.3.2 Metoda MakeHillRounded

Metoda upravuje výšku všech bodů v poloměru útvaru tak, že jejich výška nelineárně klesá směrem od středu útvaru. Tím je dosaženo zaoblení útvaru. Pro dosažení



Obrázek 4.16: Model kopce

takového tvaru autor použil funkci sinus s předpisem:

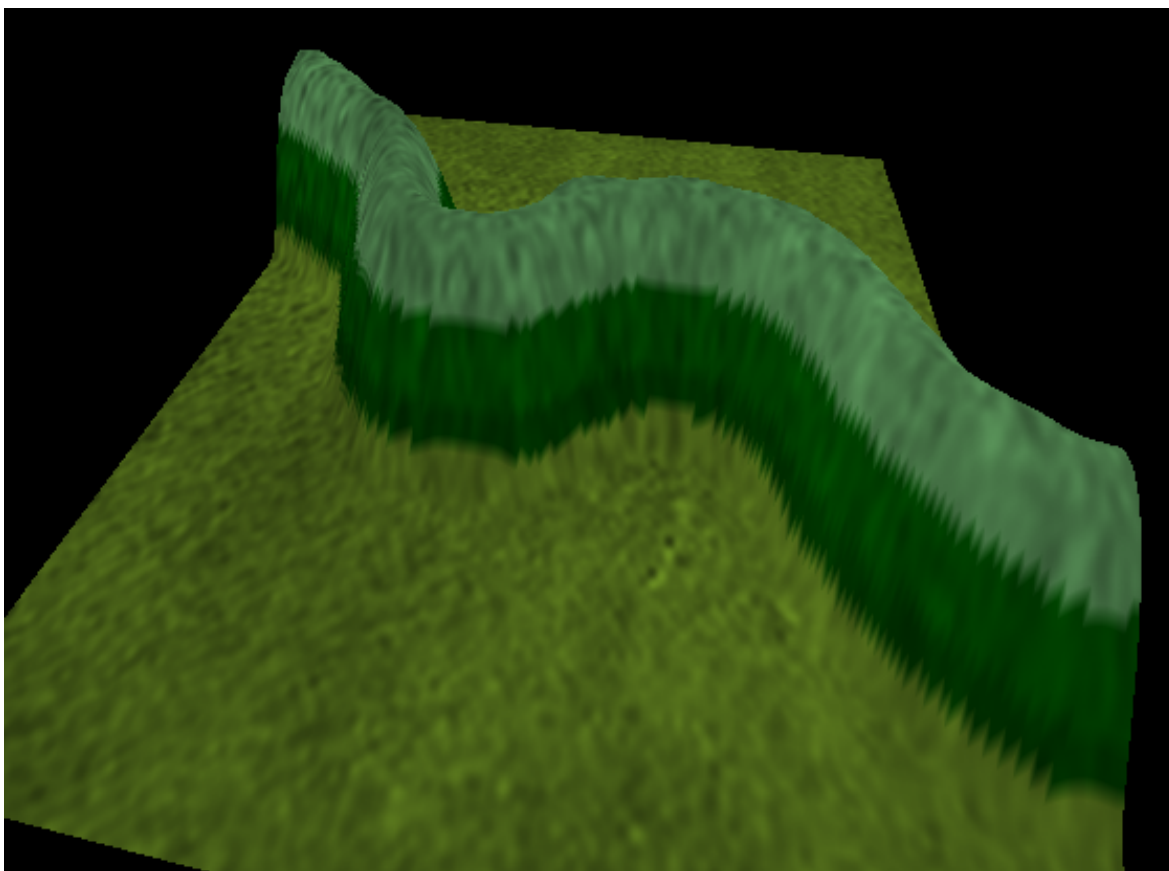
$$y = \sin\left(\frac{\lambda}{r} \cdot \frac{\pi}{2}\right) \cdot h$$

λ - vzdálenost bodu od úpatí kopce

r - poloměr kopce

h - výška kopce

Funkce generuje kopec se zaobleným vrcholem a strmým úpatím. Čím vyšší je poměr výšky kopce k jeho poloměru, tím je kopec strmější. Na obrázku 4.17 je ukázána tvorba pohoří pomocí tohoto nástroje. Vzniklý objekt je dobré pro lepší vzhled vyhladit např. 10 iteracemi algoritmu pro vyhlazování.

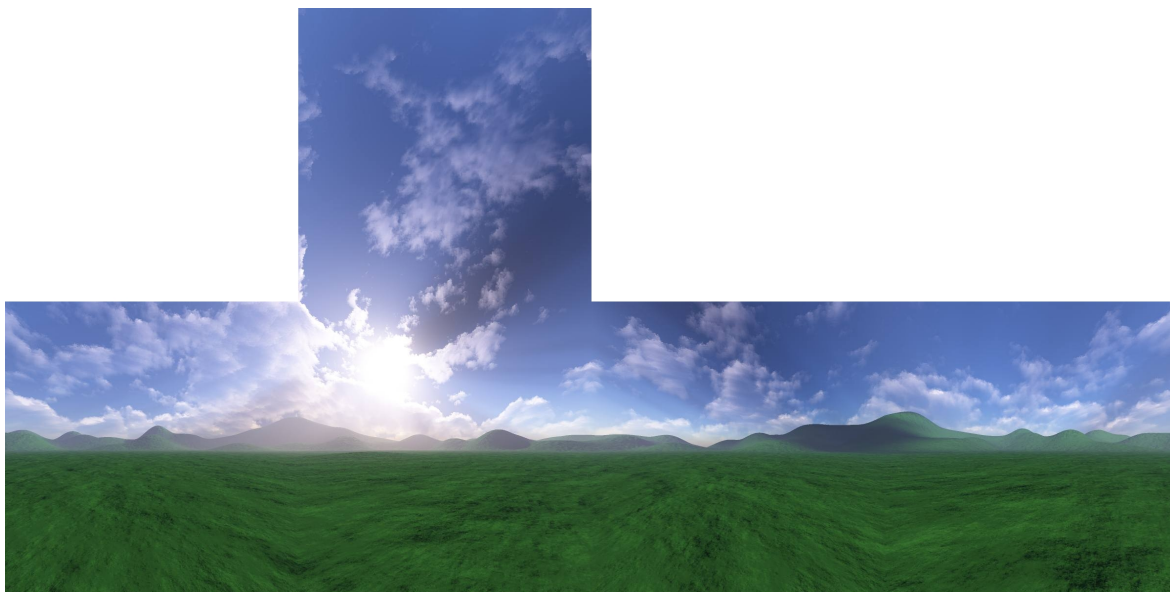


Obrázek 4.17: Pohoří

4.2.4 Skybox

V programu je autorem implementován skybox v podobě krychle. Rozměry skyboxu jsou nastaveny tak, aby obklopoval celý terén. Pozice skyboxu se nemění s pohybem kamery, jeho střed je totožný se středem terénu.

Textury lze určit pro každou stěnu skyboxu zvlášť. Aby výsledná scéna dobře vypadala, je vhodné použít kvalitní textury o vysokém rozlišení. Použité textury mají rozlišení 1024x1024 viz obr. 4.18. Snímky ze scény s využitím skyboxu jsou zobrazeny na obrázku 4.19.



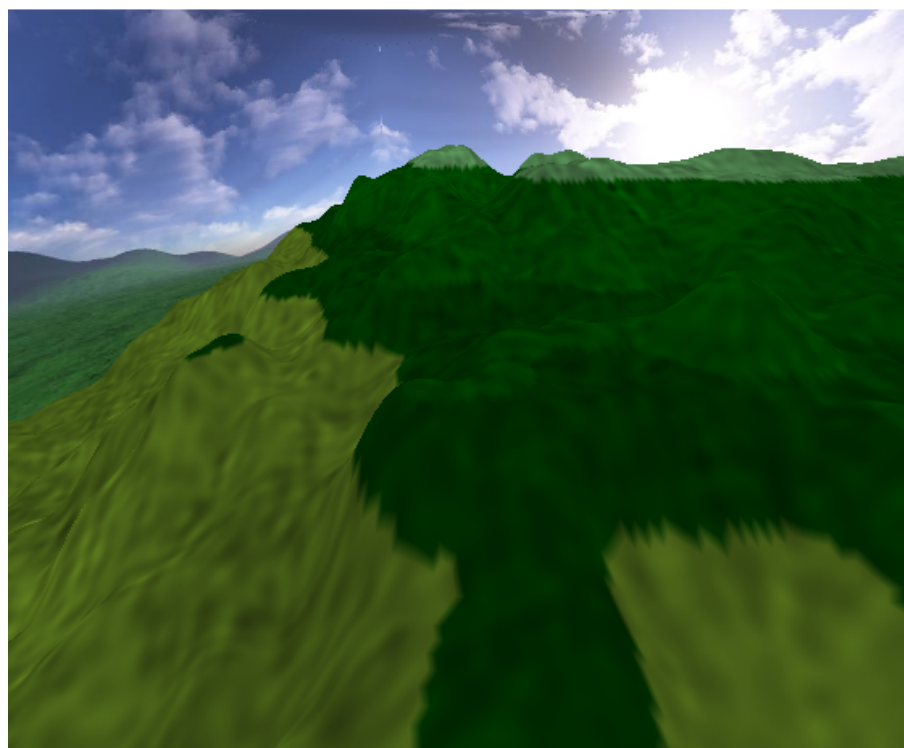
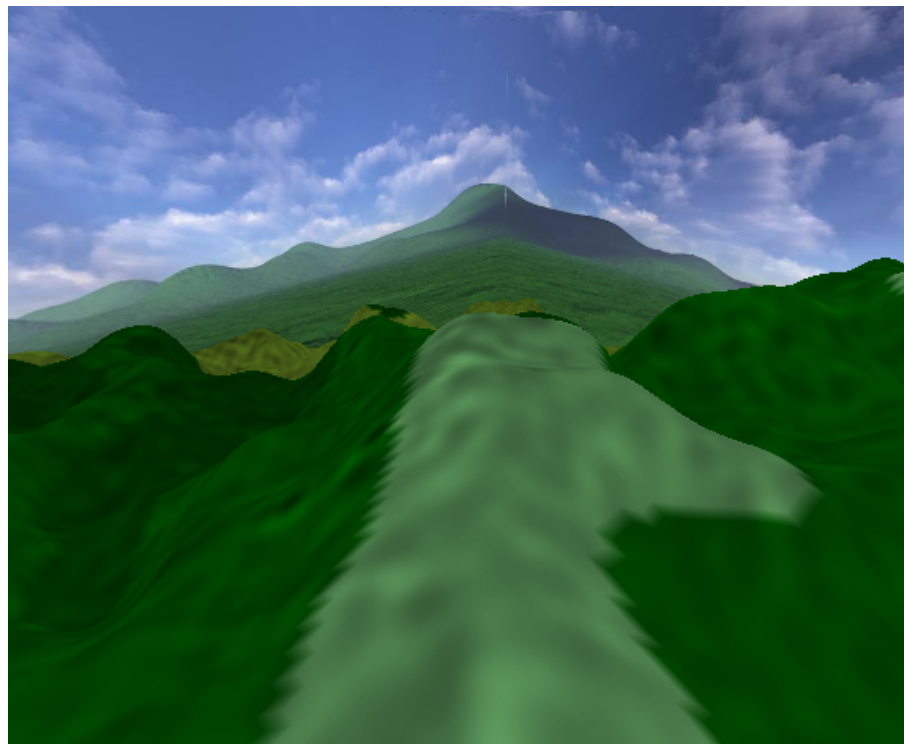
Obrázek 4.18: Síť pro skybox (převzato z <http://www.keithlantz.net/>)

4.2.5 Vodní plochy

V programu lze povolit vykreslování vodních ploch. Autorem je implementována možnost zobrazit jednu vodní plochu, jejíž rozměry (délka, šířka) jsou totožné s rozměry terénu. Vodní plocha má definováno několik vlastností, které lze přes grafické rozhraní programu měnit. Mezi vlastnosti patří barva vody, průhlednost, textura a výška vodní hladiny.

Autor použil nastavení OpenGL, které počítá průhlednost objektu v závislosti na parametru alfa. Čím je hodnota alfa nižší, tím je objekt průhlednější. Nastavení lze

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU



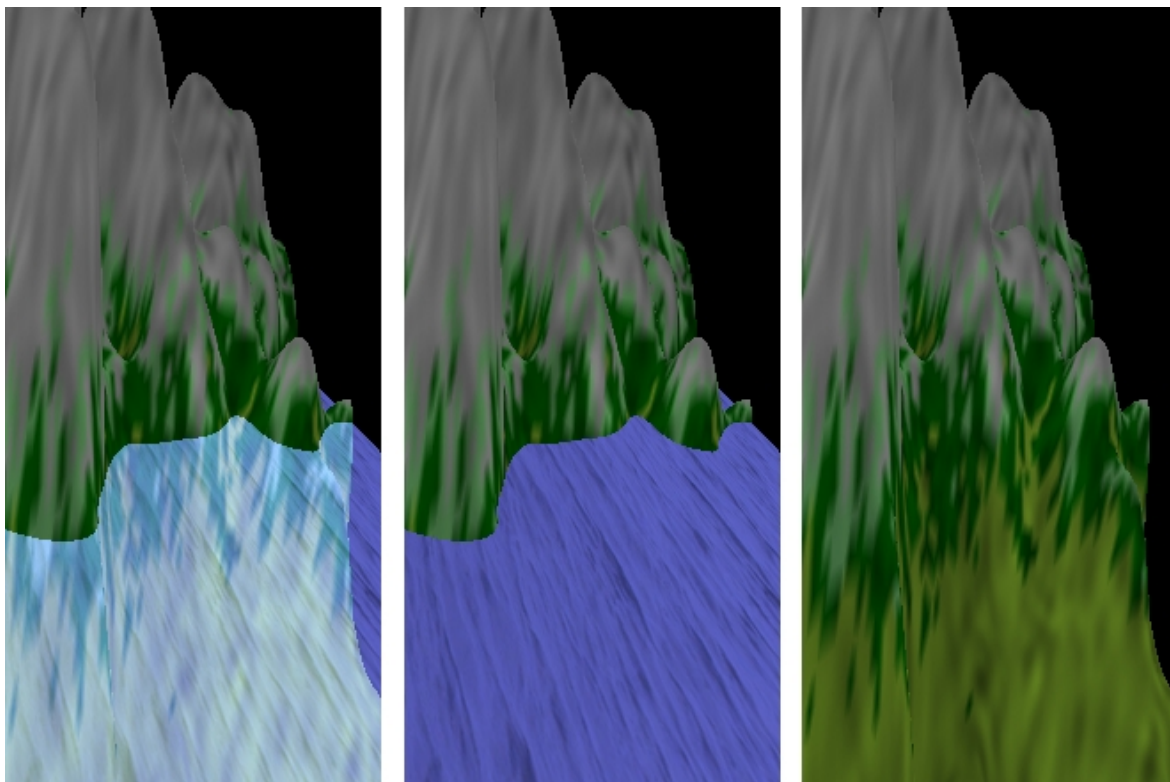
Obrázek 4.19: Náhled scény se skyboxem

KAPITOLA 4. IMPLEMENTACE PROGRAMU PRO GENEROVÁNÍ A ZOBRAZENÍ 3D TERÉNU

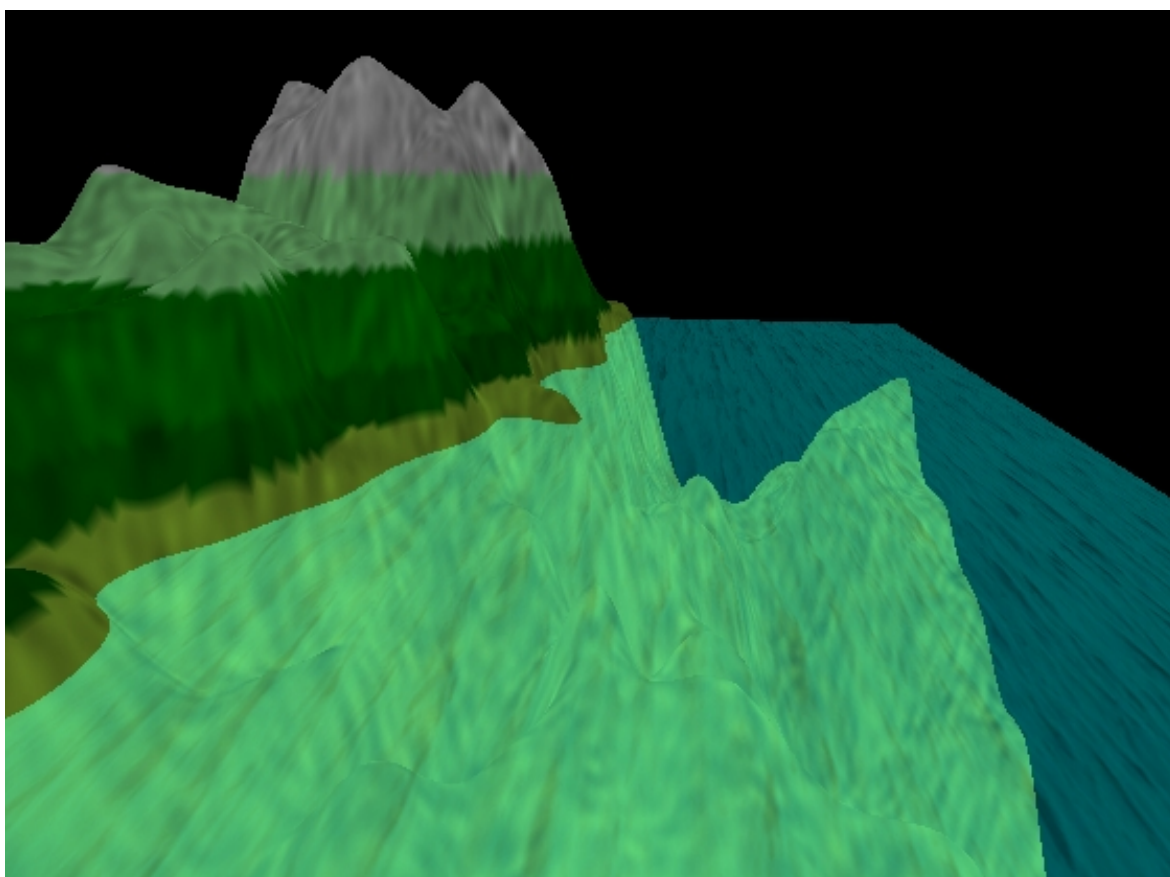
dosáhnout příkazem `glBlendFunc(GL_SRC_ALPHA, GL_ONE)`. Po použití tohoto příkazu je výsledná barva pixelu rovna součtu hodnoty cílového pixelu a hodnoty zdrojového pixelu vynásobeného hodnotou alfa. Operace zapsaná formou rovnice by měla podobu:

$$C(R, B, G) = (R_s \cdot S_a + R_d \cdot 1, G_s \cdot S_a + G_d \cdot 1, B_s \cdot S_a + B_d \cdot 1, A_s \cdot S_a + A_d \cdot 1)$$

Změna parametrů pomocí `glBlendFunc(GL_ONE, GL_ZERO)` by vedla k vykreslení vody s neprůhlednou hladinou. Naopak při nastavení hodnot na `glBlendFunc(GL_ZERO, GL_ONE)` nebudou průhledné objekty vykreslovány vůbec. Výsledky jednotlivých nastavení jsou zobrazeny na obrázku 4.20.



Obrázek 4.20: Vliv parametrů rovnice míchání na průhlednost. Zleva `glBlendFunc(GL_SRC_ALPHA, GL_ONE)`, `glBlendFunc(GL_ONE, GL_ZERO)`, `glBlendFunc(GL_ZERO, GL_ONE)`



Obrázek 4.21: Voda s texturou

4.3 Implementace

Jak již bylo zmíněno, program byl napsán v objektově orientovaném jazyku C#. Funkce programu autor rozdělil do několika tříd. Grafické rozhraní programu je realizováno přes formuláře a ostatní grafické komponenty. Pro zobrazení grafického výstupu autor použil komponentu `glControl`, která je součástí knihovny `OpenTK`.

4.3.1 Objektový model

V této části jsou stručně popsány použité třídy a jejich nejdůležitější metody.

4.3.1.1 TerrainGenerator

Třída poskytuje prostředky pro náhodné generování povrchu terénu. Na výběr jsou tři metody, každá implementuje jeden z algoritmů popsaných v kapitole 3.2.5. Metody nevracejí žádný výstup, výsledky ukládají do vnitřní proměnné, přístupné přes metodu třídy. Nyní následuje stručný popis jednotlivých metod.

- `void GenerateMidPointDisplacement`
Parametry - `int leftUpperCornerX`, `int leftUpperCornerY`, `int rightLowerCornerX`, `int rightLowerCornerY`, `float delta`, `float deltaMult`
Metoda implementuje generování terénu pomocí "přesouvání středního bodu". První čtyři parametry vymezují čtvercovou oblast, ve které části výškové mapy se bude generování odehrávat. Parametr *delta* určuje velikost změny středního bodu při každé iteraci. Parametrem *deltaMult* je po každé iteraci vynásobem parametr *delta*, čímž lze dosáhnout zvyšování resp. snižování změny středního bodu při každé další iteraci. Generování probíhá rekurzivně.
- `void GenerateParticleDeposition`
Parametry - `int numOfIterations`, `int particleEnergy`, `float energyDrain`, `int searchArea`, `bool useDiagonalNeighbours`
Metoda implementuje generování terénu pomocí "usazování částic". Parametr *numOfIterations* určuje počet částic, které budou postupně "vhozeny" do oblasti terénu, *particleEnergy* určuje energii částice, *energyDrain* udává pokles energie částice po každé iteraci, parametr *searchArea* definuje do jaké vzdálenosti od aktuální pozice se může částice přesunout a nastavení *useDiagonalNeighbours* na

hodnotu false zajistí pouze vertikální, nebo horizontální přesun částice.

- void GenerateFaultFormation

Parametry - int numOfIterations, float delta, float deltaMult

Metoda implementuje generování terénu pomocí "náhodných poruch". Za objekt, rozdělující rovinu na dvě části byla zvolena náhodně umístěná přímka. Parametry mají stejný význam jako příslušné parametry u předchozích dvou metod.

4.3.1.2 SkyBox

Třída poskytuje metody pro zobrazení skyboxu. Skybox je implementován jako kvádr, obklopující celý zobrazený terén. Každé ze stěn lze přiřadit texturu. Třída poskytuje metody pro úpravu pozice, rozměrů (šířka, výška, délka), textury, a funkci (void Draw()) pro vykreslení.

4.3.1.3 Water

Třída poskytuje metody pro zobrazení vodních ploch. Voda je implementována jako čtverec vykreslený v zadané výšce, zobrazený přes celou oblast terénu. Třída obsahuje atributy pro nastavení konstant zdrojového a cílového faktoru pro výpočet průhlednosti, barvy a textury, a funkci (void Draw()) pro vykreslení.

4.3.1.4 Terrain

Třída poskytuje metody a atributy pro práci s modelem terénu. Model je zobrazován ve formě pravidelné trojúhelníkové sítě. Referenční body jsou uloženy v dvourozměrném poli reálných čísel, datový typ double, Data. Atribut ColorData, dvourozměrné pole barev, uchovává barvy pro jednotlivé referenční body. Atribut ColorData umožňuje barevně odlišit referenční body v závislosti na jejich výšce, proto je nutné při každé změně výšky referenčního bodu volat funkci ComputeColorData(), která provede aktualizaci pole barev podle výšky referenčních bodů. Pro renderování skyboxu a vodní hladiny obsahuje třída objekty typu SkyBox a Water. Třída dále poskytuje metody pro uložení a načtení modelu ze souboru SaveToFile(string path) a LoadFromFile(string path).

- void ComputeColorData()

Parametry: bez parametrů nebo Color color

Provede aktualizaci pole ColorData v závislosti na hodnotách v poli Data. Pokud

je metoda volána bez parametru, jsou určitým rozmezím výšky přiřazeny různé barvy. Při volání metody s parametrem, je všem bodům přiřazena barva uvedená v parametru.

- Smooth()

Parametry: int Passes

Metoda implementuje algoritmus vyhlazování uvedený v kapitole 4.2.2. Parametr *Passes* udává počet iterací algoritmu, které se na terén aplikují.

- Draw()

Parametry: bez parametrů

Vykreslí model terénu v závislosti na hodnotách vnitřních proměnných. Povrch terénu může být buďto pokryt pouze barvou, pokryt pouze texturou nebo pokryt barvou a texturou, přičemž výsledná barva je určena funkcí `GL_TexEnv()` volanou popořadě s parametry `GL_TEXTURE_ENV`, `GL_TEXTURE_ENV_MODE`, `GL_MODULATE`, která vypočte výslednou barvu vynásobením barvy pixelu a k němu příslušnému texelu fragmentu. Mimo vykreslení modelu terénu také volá metody pro vykreslení skyboxu (`SkyBox.Draw()`) a vodní hladiny (`Water.Draw()`).

- MakeHill()

Parametry: int X, int Y, float height, int radius, HillType hillType, bool additive = false

Účelem metody je tvorba kopců viz kapitola 4.2.3. Parametry *x* a *y* určují pozici vrcholu kopce. *Height* určuje výšku kopce v jeho nejvyšším místě, *radius* určuje poloměr kopce, parametrem *hillType* lze specifikovat typ generovaného útvaru a parametr *additive*, pokud je nastaven na false, přepisuje pouze ty hodnoty referenčních bodů, které mají nižší výšku než je hodnota parametru *height*.

Kapitola 5

Závěr

Cílem práce bylo seznámit se s problematikou počítačového modelování se zaměřením na modelování terénu a navrhnout aplikaci pro generování modelů terénu.

Cílem literární rešerše bylo prozkoumat metody a postupy používané při tvorbě digitálních modelů terénu. Nejdříve byly popsány obecné postupy počítačového modelování. Dále byly probrány možné zdroje dat pro digitální modely terénu a datové struktury, do kterých mohou být tato data uložena. V některých případech je potřeba z již naměřených dat odvodit data nová. Tyto postupy byly probrány v části zabývající se interpolací dat. V závěru teoretické části práce byly popsány metody týkající se problematiky zobrazení modelu terénu a některé vlastnosti OpenGL, které byly využity při praktické části práce.

V praktické části práce byla implementována aplikace pro zobrazení, generování a úpravy modelů terénu. V aplikaci byly implementovány různé algoritmy pro náhodné generování terénů. Jejich průběh lze ovlivňovat změnou počátečních parametrů. Stejně tak byly implementovány funkce pro ruční úpravu modelu. Dále byly do programu zahrnuty funkce pro renderování vodní hladiny, u které byla implementována možnost nastavení různých parametrů, které ovlivňují její výsledný vzhled. Stejně tak byla zahrnuta i možnost zobrazení skyboxu. Pro zvýšení věrohodnosti vygenerovaného modelu byly do programu implementovány funkce pro vyhlazování. Dále byla přidána komponenta pro náhled vygenerovaného modelu terénu.

Autor stanovený cíl splnil a na základě technik a metod popsaných v teoretické části práce vytvořil aplikaci umožňující generovat, upravovat a zobrazovat modely terénu.

Kapitola 6

Slovníček pojmů

V této části jsou vysvětleny vybrané termíny a zkratky použité v praktické a teoretické části práce.

API (Application Programming Interface) Sada funkcí určená pro komunikaci softwarových komponent mezi sebou.

CSG (Constructive Solid Geometry) Způsob modelování tělesa, při kterém výsledný objekt vznikne složením z jednodušších objektů.

Buffer Oblast paměti pro dočasné uložení dat.

Fragment Pixel geometrického primitiva ve framebufferu.

Framebuffer Skupina bufferů, z nichž každý je určen pro jednu nebo více operací, které lze s daty při vykreslování provádět.

Drátěný (wireframe) model Taková reprezentace, ve které jsou zobrazeny pouze hrany spojující jednotlivé vrcholy objektu.

Shader Program spuštěný a vykonávaný procesorem na grafické kartě. Pokud je program aktivní, ovlivňuje všechny vrcholy (vertex-shader) nebo fragmenty (fragment-shader, pixel-shader; fragment), které grafická karta zpracovává.

Smooth surface Povrch je hladký, pokud funkce, která jej popisuje, je spojitá ve své první a následujících derivacích.

Surface reconstruction (surface modelling) Proces tvorby povrchu terénu. Jeho výsledkem je digitální model.

Texel (Texture element) rastrový element textury.

Literatura

- [1] ŽÁRA, Jiří, et al. *Moderní počítačová grafika*. 2. vyd. Brno: Computer Press, 2004. 609 s. ISBN 80-251-0454-0, K1143.
- [2] SHREINER, Dave, et al. *OpenGL : Průvodce programátora*. 1. vyd. Brno : Computer Press, 2006. 679 s. ISBN 80-251-1275-6, K1198.
- [3] FOLEY, James D., et al. *Computer Graphics : Principles and Practice*. 2. vyd. [USA] : Addison Wesley, 1995. 1200 s. ISBN 0201848406.
- [4] POLACK, Trent. *Focus on 3D Terrain Programming*. 1. vyd. Cincinnati (Ohio) : Premier Press, 2003. 218 s. ISBN 1-59200-028-2.
- [5] HEARN, Donald; BAKER, M. Pauline. *Computer graphics : C version* 2. vyd. [USA] : Prentice Hall Press, 1997. 652 s. ISBN 0135309247.
- [6] OLSEN, Jacob. *Realtime Procedural Terrain Generation*. University of Southern Denmark, 31.10.2004. Dostupné z: <http://web.mit.edu/cesium/Public/terrain.pdf>. Pojednání. University of Southern Denmark.
- [7] LI, Zhilin; ZHU, Qing; GOLD, Christopher. *Digital Terrain Modeling: Principles and methodology* [USA] : CRC Press, 2005. 318 s. ISBN 0-415-32462-9
- [8] EL-SHEIMY, Naser; VALEO, Caterina; HABIB, Ayman. *Digital Terrain Modeling : Acquisition, manipulation and applications* [USA]: Artech house, 2005. 257 s. ISBN 1-58053-921-1
- [9] TIŠNOVSKÝ, Pavel. *Root* [online]. 11. 3. 2008 [cit. 2011-12-18]. Konstruktivní geometrie těles (CSG). Dostupné z WWW: <<http://www.root.cz/clanky/konstruktivni-geometrie-teles-csg/>>

LITERATURA

- [10] WOJCIECH, Toman. *Wojciech Toman's devlog* [online]. 12. 5. 2009 [cit. 2012-2-2]. Procedural terrain generation - particle deposition. Dostupné z WWW <<http://wtomandev.blogspot.com/2009/05/procedural-terrain-generation-particle.html>>
- [11] TUREK, Michal. *CZ NeHe OpenGL* [online]. 23.07.2004 [cit. 2012-2-5] Octree. Dostupné z WWW: <http://nehe.ceske-hry.cz/cl_gl.octree.php>
- [12] FOSTER, Nic. *Nic's GameDev Site* [online]. 27.09.2007 [cit. 2012-3-1] Heightmap Terrain Smoothing. Dostupné z WWW: <<http://nfostergames.com/Lessons/TerrainSmoothing.htm>>

Seznam obrázků

3.1	Ukázka částicového systému	8
3.2	CSG strom	9
3.3	Princip metody přesouvání středního bodu	13
3.4	Přesouvání středního bodu	13
3.5	Metoda náhodných poruch	14
3.6	Usazování částic	15
3.7	Point-based model	17
3.8	Lineární interpolace	21
3.9	Bilineární interpolace	22
3.10	Dědění vlastností v grafu scény	26
3.11	Čtyřstrom	28
3.12	Skybox: použité textury a výsledná scéna	30
4.1	Nastavení vykreslování	36
4.2	Nastavení generátoru	37
4.3	Nastavení štětce	37
4.4	Nastavení vodní hladiny	38
4.5	Nastavení skyboxu	39
4.6	Nastavení vyhlazování	39
4.7	Přesun středního bodu - model č. 1, vpravo po vyhlazení	41
4.8	Přesun středního bodu - model č. 2, vpravo po vyhlazení	42
4.9	Metoda náhodných poruch - model č. 1, vpravo po vyhlazení	44
4.10	Metoda náhodných poruch - model č. 2, vpravo po vyhlazení	44
4.11	Metoda náhodných poruch - model č. 3, vpravo po vyhlazení	45
4.12	Metoda náhodných poruch - model č. 4, vpravo po vyhlazení	46
4.13	Metoda usazování částic - model č. 1, vpravo po vyhlazení	48

4.14	Metoda usazování částic - model č. 2, vpravo po vyhlazení	49
4.15	Proces vyhlazování po 0, 10, 20, 30 a 40 iteracích	51
4.16	Model kopce	52
4.17	Pohoří	53
4.18	Síť pro skybox	54
4.19	Náhled scény se skyboxem	55
4.20	Vliv parametrů rovnice míchání na průhlednost	56
4.21	Voda s texturou	57

Seznam tabulek

4.1	Metoda přesunu středního bodu - parametry modelů	40
4.2	Metoda náhodných poruch - parametry modelů	43
4.3	Metoda usazování částic - parametry modelů	47