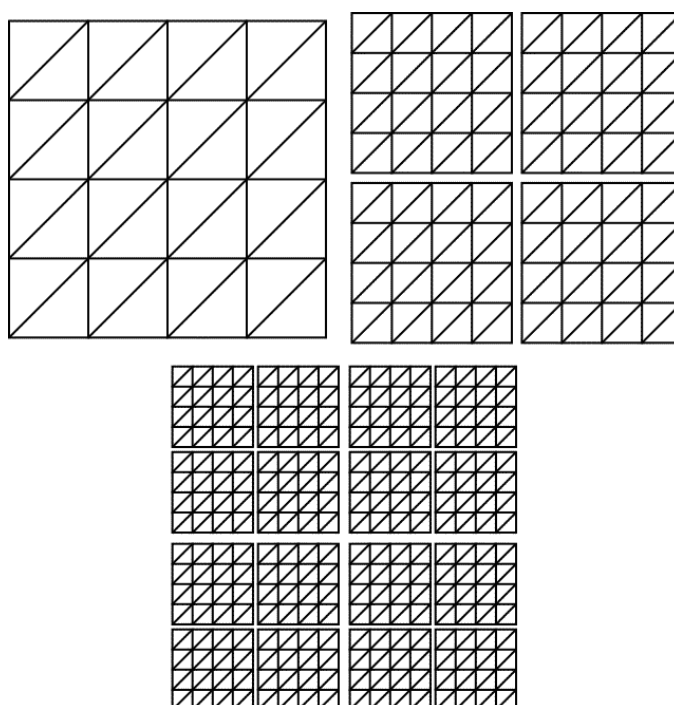


Chunked LoD algoritmus

1. Motivace:

Od roku 2001 vyvíjí Ulrich Thatcher svou vlastní alternativu k známým algoritmům LoD terénu jakými jsou např. ROAM [2], SOAR [3] a Geo Mip-Mapping [4] nazvanou Chunked LoD [1].

Hlavním principem tohoto algoritmu je reprezentace zobrazovaného terénu jako kvadrantového stromu dlaždic na různých úrovních detailů a vybírání zobrazovaných dlaždic podle aktuální pozice kamery vzhledem k terénu. Dlaždice jsou ve stromu hierarchicky organizovány tak, že kořenem stromu je dlaždice pokrývající celý zobrazovaný terén s nejnižší úrovní detailů a ta je pak rekurzivně dělena na čtyři dlaždice s dvojnásobnou úrovní detailů. První tři úrovně takového stromu jsou zobrazeny na obrázku 1.



Obrázek 1: První tři úrovně kvadrantového stromu dlaždic.

Vzhledem k tomu, že při záměně rodičovské dlaždice za čtyři potomky za běhu algoritmu dochází k relativně náhlé změně geometrie triangulace, je nutno plynule přizpůsobovat geometrii dlaždic s vyšší úrovní detailů geometrii dlaždice s úrovní detailů nižší. Tato technologie se v anglické terminologii nazývá geomorphing a v našem případě jí lze dosáhnout interpolací výšky bodů na sudých řádcích a sloupcích dlaždice s průměrnou výškou bodů na lichých řádcích nebo sloupcích dlaždice. Za předpokladu, že chyba zobrazení rodičovské dlaždice je dvakrát vyšší než chyba zobrazení dlaždic potomků, můžeme snadno vypočíst podle příslušných vzorců uvedených v [1] faktor pro tuto interpolaci. V praxi to pak znamená, že těsně po změně dlaždice s nižší úrovní detailů za dlaždici s vyšší úrovní detailů je faktor roven 0 a její geometrie je podobná dlaždici původní. Blíží-li se kamera blíže k dlaždici, geometrie se postupně mění na geometrii dané dlaždice. Když faktor dosáhne hodnoty 1 má již dlaždice svoji geometrii úplně.

2. Implementace:

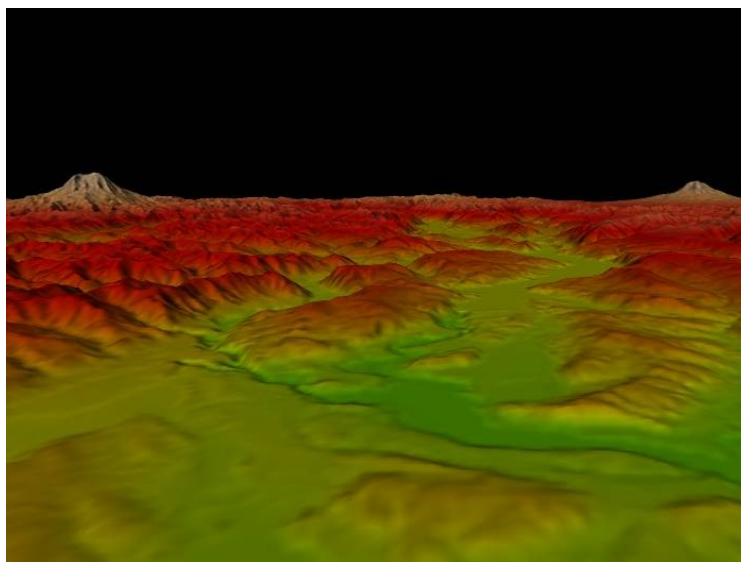
Tento projekt tvoří součást knihovny SoTerrain [5], která je rozšířením knihovny Coin [6] o sadu uzlů grafu scény implementujících algoritmy vizualizace terénu a která byla hlavní součástí mé bakalářské práce. V rámci tohoto projektu byly implementovány třídy SoSimpleChunkedLoDTerrain, SbChunkedLoDTileTree a SbChunkedLoDTile a změněn sestavovací systém z obyčejného Makefile na

systém GNU Automake [7].

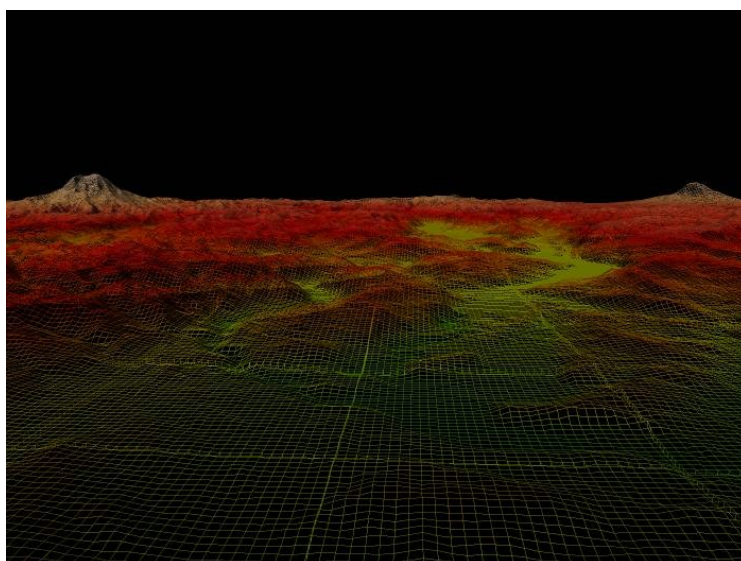
Uzel grafu scény `SoSimpleChunkedLoDTerrain` se používá tak, že se mu předřadí uzly tříd `SoCoordinate3`, `SoTexture2`, `SoTextureCoordinate2`, `SoNormal` a `SoNormalBinding` obsahující souřadnice bodů zobrazované výškové mapy, texturu, texturové souřadnice, normály a způsob použití normál. Uzel terénu pak v prvním snímku získá informace z těchto uzlů pomocí elementů a vytvoří si kvadrantový strom dlaždic. Protože aktuální implementace používá pravidelné dělení dlaždice na trojúhelníky (takové jako je na obrázku 1) místo adaptivního dělení uvedeného v dokumentu [1], je poněkud složitější dosáhnout pravidla, že chyba zobrazení rodičovské dlaždice je dvojnásobek chyby zobrazení dlaždic potomků, a složitější je i následný výpočet interpolačního faktoru. Toto je řešeno ořezáním výsledného faktoru do rozsahu $<0, 1>$. Je-li faktor roven 1 neuplatňuje se interpolace vůbec, a vykreslovací funkce je nahrazena optimálnější. Dále, aby tato funkce byla používána častěji, je faktor ještě před ořezáním vynásoben konstantou 2, aby bylo konečné geometrie dlaždice dosaženo dříve.

Pokud je to povoleno, probíhá také ořezávání dlaždic pohledovým tělesem. To znamená, že pokud kvádr ohraničující každou dlaždicí je mimo obrazovku, je dlaždice vyloučena z vykreslování dříve, než jsou její trojúhelníky poslány na grafickou kartu. Toto velice výrazně ovlivňuje celkovou výkonnost algoritmu.

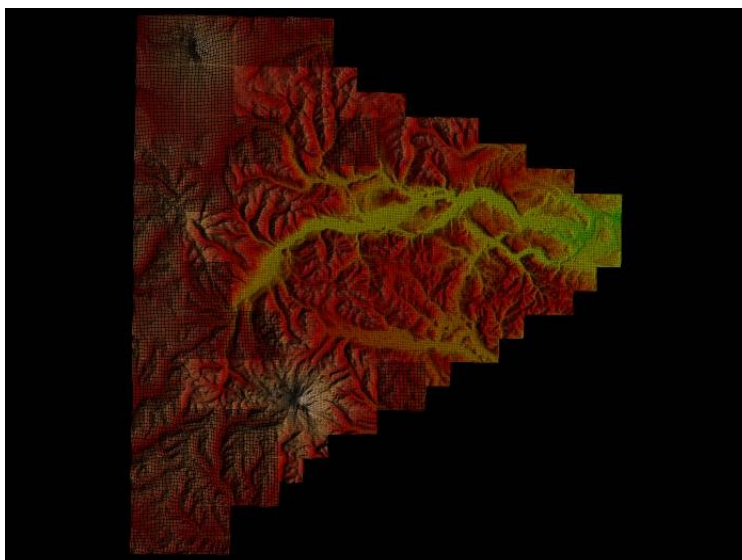
Dále následují tři obrázky demonstrující průběh algoritmu, první dva zobrazují terén vykreslený algoritmem `Chunked LoD` plnými trojúhelníky resp. jako drátový model, poslední obrázek ukazuje důsledek ořezávání pohledovým tělesem.



Obrázek 2: Terén zobrazený algoritmem `Chunked LoD` plnými trojúhelníky.



Obrázek 3: Terén zobrazený algoritmem `Chunked LoD` jako drátový model.



Obrázek 4: Demonstrace účinku ořezávání pohledovým tělesem algoritmu Chunked LoD.

3. Použití testovací aplikace:

Pro demonstraci implementovaných algoritmů knihovny SoTerrain vznikla jednoduchá aplikace SoTerrainTest, kterou lze přeložit příkazem `make` v hlavním adresáři. Nejprve je však nutné nakonfigurovat knihovnu pomocí skriptů `./autogen.sh` a `./configure`. Při této konfiguraci lze specifikovat, zda má být knihovna přeložena s debugovacími nebo profilovacími informacemi či s vlastní knihovnou pro profilování. Seznam těchto a dalších možností včetně nápovědy lze zobrazit příkazem `./configure --help`. Popis, jak přeložit a spustit projekt na počítači `merlin.fit.vutbr.cz` je v souboru `README`.

Parametry a použití testovací aplikace je následující:

```
Použití: SoTerrainTest -h heightmap [-t texture] [-p profile_file]
        [-a algorithm] [-A animation_time] [-F frame_time] [-e pixel_error]
        [-r triangle_count] [-g tile_size] [-f] [-c] [-v] [-s]
        -h heightmap           Soubor se vstupní výškovou mapou.
        -t texture             Soubor s texturou terénu.
        -p profile_file        Soubor pro uložení výsledku profilování.
        -a algorithm           Výběr algoritmu pro zobrazení terénu.
                               brutalforce   Vykreslení terénu hrubou silou.
                               roam         Vykreslení terénu algoritmem ROAM.
                               geomipmapping Vykreslení terénu algoritmem Geo Mip-Mapping.
                               chunkedlod   Vykreslení terénu algoritmem Chunked LoD.
        -A animation_time      Délka animace v milisekundách.
        -F frame_time          Doba trvání jednoho snímku v milisekundách.
        -e pixel_error         Nastavení chyby zobrazení algoritmu.
        -r triangle_count      Nastavení počtu zobrazovaných trojúhelníků.
        -g tile_size           Nastavení velikosti dlaždice.
        -f                     Spuštění programu na celé obrazovce.
        -c                     Zapnutí ořezávání pohledovým tělesem.
        -v                     Spuštění animace po startu programu.
        -s                     Zapnutí synchronizace snímků animace.
```

Parametr `-r` funguje, pouze je-li zvolen algoritmus ROAM, a parametr `-g` funguje jenom u algoritmu Geo Mip-Mapping a Chunked LoD. Je-li zvolen parametr `-s`, trvá animace přesně dobu nastavenou parametrem `-A`, v opačném případě může dojít ke zpoždování vykreslování snímků a animace se prodlouží. Ukládání výsledků profilování probíhá pouze tehdy, je-li knihovna nakonfigurována s parametrem `--with-profile`. Není-li zadán parametr `-v`, je zobrazování terénu interaktivní s následujícím významem kláves klávesnice (Pro názvy kláves byly použity názvy běžné pro knihovnu Coin):

- W – posun kamery kupředu
- S – posun kamery dozadu
- A – posun kamery doleva
- D – posun kamery doprava

Q – posun kamery nahoru
E – posun kamery dolů
LEFT – natočení kamery doleva
RIGHT – natočení kamery doprava
UP – natočení kamery nahoru
DOWN – natočení kamery dolů
PGUP – naklonění kamery doleva
PGDOWN – naklonění kamery doprava
PAD ADD – zvýšení chyby zobrazení v pixelech
PAD SUBTRACT – snížení chyby zobrazení v pixelech
F – zapnutí/vypnutí ořezávání pohledovým tělesem
L – zapnutí/vypnutí zobrazení terénu pomocí úseček
LEFT ALT+ENTER – přepnutí celoobrazovkového režimu prohlížeče
ESC – ukončení aplikace

4. Použité zdroje:

- [1] URLICH, Thatcher. Rendering Massive Terrains using Chunked Level of Detail Control. Oddworld Inhabitants, 2002. 14 stran. Dostupný z: http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/*checkout*/tu-testbed/tu-testbed/docs/sig-notes.pdf?rev=HEAD
- [2] DUCHAINEAU, Mark A., WOLINKSY, Murray, SIGETI, David E. MILLER, Mark C., ALDRICH, Charles a MINEEV-WEINSTEIN, Mark B. ROAMing terrain: Real-time optimally adapting meshes. IEEE Visualization '97. 1997, strany 81—88. Dostupný z: <http://www.llnl.gov/graphics/ROAM/roam.pdf>
- [3] LINDSTROM Peter a PASCUCCI, Valerio. Terrain Simplification Simplified: A General Framework for View-Dependent Out-of-Core Visualization. IEEE Transactions on Visualization and Computer Graphics. 2002, roč. 8, č. 3, strany 239–254. Dostupný z: <http://www.pascucci.org/pdf-papers/IEEE-TVG-2002.pdf>
- [4] DE BOER, Willem H. Fast Terrain Rendering Using Geometrical MipMapping. 2000. 7 stran. Dostupný z: http://www.flipcode.com/articles/article_geomipmaps.pdf
- [5] Knihovna algoritmů pro vizualizaci terénu SoTerrain: <http://blackhex.no-ip.org/wiki/SoTerrain>
- [6] Knihovna Coin, implementace knihovny OpenInventor: <http://coin3d.org>
- [7] Systém GNU Automake: <http://sources.redhat.com/automake/>