



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ZÁSUVNÝ MODUL PRO BLENDER PRO EROZI

BLENDER PLUGIN FOR EROSION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETER URGOŠ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. TOMÁŠ MILET, Ph.D.

BRNO 2022

Zadání bakalářské práce



Student: **Urgoš Peter**
Program: Informační technologie
Název: **Zásuvný modul pro Blender pro erozi**
Blender Plugin for Erosion
Kategorie: Počítačová grafika

Zadání:

1. Nastudujte si editor Blender a Blender API. Nastudujte si techniky eroze. Vyhledejte stávající řešení a zhodnoťte jejich vlastnosti.
2. Navrhněte plugin pro editor Blender, který umožňuje erozi modelů.
3. Implementujte navržený plugin.
4. Uživatelsky otestujte a zhodnoťte výsledky.
5. Vytvořte demonstrační video.

Literatura:

- Blender Python API Documentation. <https://docs.blender.org/api/current/index.html>
- Stava, Ondrej, et al.: Interactive Terrain Modeling Using Hydraulic Erosion. Eurographics/SIGGRAPH Symposium on Computer Animation (2008)
- Julie Dorsey, et al.: Modeling and Rendering of Weathered Stone. In Proceedings of SIGGRAPH '99, ACM Press / ACM SIGGRAPH (1999)

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2 a kostra pluginu.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Milet Tomáš, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 1. listopadu 2021

Abstrakt

Táto bakalárska práca sa zaoberá eróziou terénu. Cieľom je navrhnúť a implementovať zásuvný modul pre 3D grafický editor Blender, umožňujúci eróziu objektov. Vytvorený nástroj ponúka metódy termálnej a vodnej erózie s užívateľsky nastaviteľnými parametrami simulácie.

Abstract

This bachelor thesis covers the topic of terrain erosion. The goal is to design and implement add-on for 3D graphics editor Blender, enabling erosion of objects. Resulting tool offers methods of thermal and hydraulic erosion with user configurable parameters of the simulation.

Klíčová slova

Blender, zásuvný modul, add-on, erózia, terén, erózia terénu, termálna erózia, vodná erózia, Python

Keywords

Blender, plugin, add-on, erosion, terrain, terrain erosion, thermal erosion, hydraulic erosion, Python

Citace

URGOŠ, Peter. *Zásuvný modul pro Blender pro erozi*. Brno, 2022. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Milet, Ph.D.

Zásuvný modul pro Blender pro erozi

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Tomáša Mileta, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Peter Uργοš
18. května 2022

Poděkování

Rád by som podakoval pánovi Ing. Tomášovi Miletovi, Ph.D za jeho odbornú pomoc pri tvorbe tejto práce.

Obsah

1	Úvod	2
1.1	Vytvorený zásuvný modul	3
2	Stav existujúcich riešení	4
2.1	Terragen	4
2.2	Unreal Engine	5
2.3	Unity	6
2.4	Blender	7
3	Metódy erózie	9
3.1	Celulárny automat	9
3.2	Termálna erózia	9
3.3	Vodná erózia	11
4	Návrh zásuvného modulu pre eróziu	16
5	Implementácia	18
5.1	Použité technológie a nástroje	18
5.2	Zásuvný modul pre Blender	20
5.3	Užívateľské rozhranie	21
5.4	Vytváranie terénu	24
5.5	Termálna erózia	27
5.6	Vodná erózia pomocou celulárneho automatu	28
5.7	Vodná erózia pomocou simulácie častíc	28
5.8	Zobrazenie postupu erózie	30
5.9	Porovnanie výsledkov implementovaných metód	31
6	Testovanie	35
7	Záver	36
	Literatura	37

Kapitola 1

Úvod

Táto bakalárska práca sa zaoberá eróziou terénu. Touto témou sa výskumníci z oblasti počítačovej grafiky zaoberajú od konca 80 rokov minulého storočia. Výpočetná kapacita počítačov za toto obdobie rapídne vzrástla a jej využitie sa našlo v mnohých oblastiach. Moderné videohry svojou realistickou grafikou ohromujú hráčov, so svetmi ktoré sa od skutočných líšia len málo. Filmové štúdiá čím ďalej častejšie siahajú po scénach „na zelenom plátne“, ktoré šetria financie a čas.

V oboch oblastiach sa tvorcovia snažia dosiahnuť realistické prostredie využitím počítačovej grafiky. Jedným z obľúbených spôsobov vytvárania terénu, ktorý je základom takmer každej outdoorovej scény, je procedurálne generovanie pomocou šumu. Táto metóda produkuje nekonečné terény, ktoré avšak nepôsobia veľmi realisticky. Reálny terén konštantne podlieha vplyvom erózie rôznych pôvodov, vietor, voda, teplota, gravitácia a mnohé iné faktory vplývajú na terén a tvarujú ho. Simulácia erózie dokáže dodať aj umelo vyzerajúcemu terénu rysy reálnej scenérie a tak posunúť realistickosť na vyššiu úroveň.



Obrázek 1.1: Snímok z videohry *Witcher 3* zobrazujúci realistický terén so známami erózie. Prevzaté z cardiffstudentmedia.co.uk.



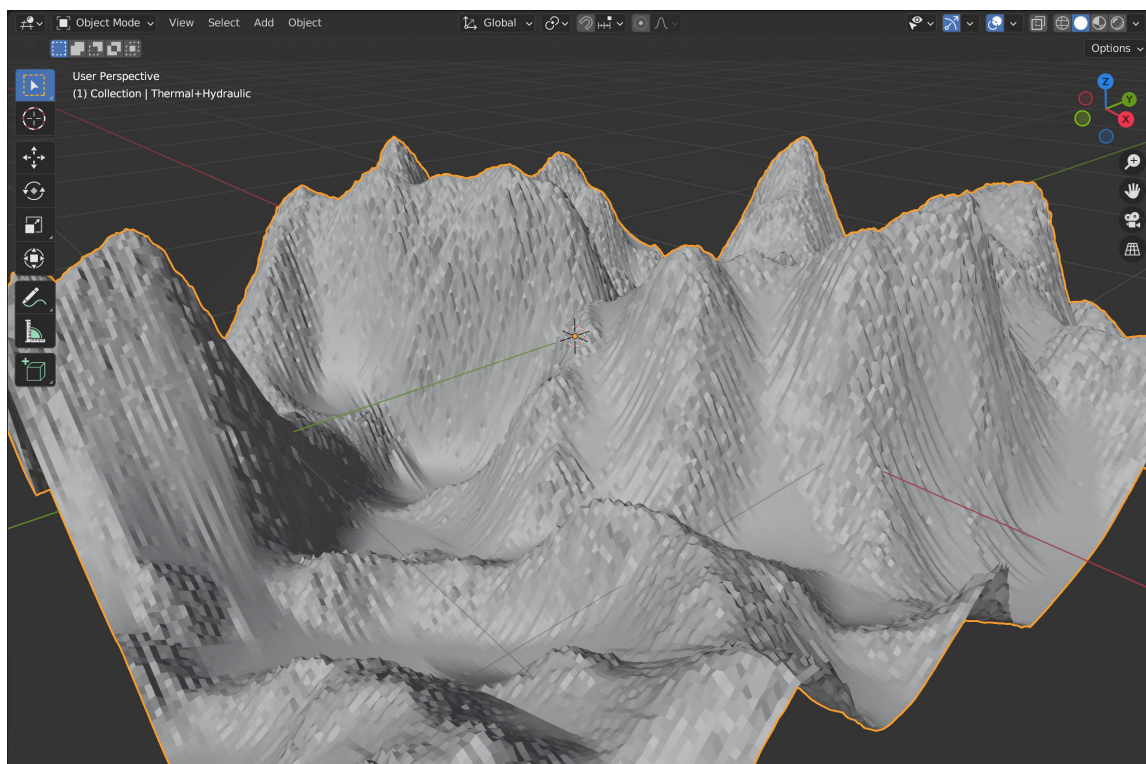
Obrázek 1.2: Paramount logo vytvorené v nástroji Terragen. Zobrazený vrch ukazuje rysy vodnej erózie. Prevzaté z wallpaperflare.com.

Cielom tejto práce je navrhnúť a implementovať zásuvný modul pre Blender, ktorý umožní simulovať eróziu objektov. Blender (kapitola 2.4) je bezplatný nástroj pre prácu s 3D grafikou, s rastúcim počtom užívateľov a funkcionalitou porovnateľnou s komerčnými produktami. Obsahuje nástroje potrebné pre tvorbu virtuálnych scén, vrátane terénu, avšak chýba mu možnosť erózie terénu. Preto sa často pre tvorbu terénu využívajú iné programy,

ako napr. Terragen (kapitola 2.1), ktorý používajú aj veľké filmové štúdiá na tvorbu realistického terénu. Populárne herné engine ako Unity (kapitola 2.3) a Unreal Engine (kapitola 2.2) taktiež ponúkajú nástroje pre eróziu terénu. Vytvorený zásuvný modul zjednodušuje tvorbu realistického terénu priamo v Blenderi.

1.1 Vytvorený zásuvný modul

Vytvorený zásuvný modul umožňuje eróziu terénu pomocou termálnej a vodnej erózie. Užívateľ dokáže upravovať nastavenia simulácie v grafickom rozhraní modulu. Taktiež umožňuje vytvárať procedurálne generovaný terén.



Obrázek 1.3: Ukážka procedurálne vygenerovaného a zerodovaného terénu pomocou vytvoreného zásuvného modulu.

Kapitola 2

Stav existujúcich riešení

Erózia terénu nie je nová téma a existuje niekoľko programov ktoré sa ňou zaoberajú. medzi najznámejšie nástroje na vytváranie terénu patrí Terragen (kapitola 2.1), ktorý obsahuje aj nástroj na eróziu terénu.

Nástroje na eróziu tiež ponúkajú pokročilé herné enginy, napríklad Unreal Engine (kapitola 2.2), alebo Unity (kapitola 2.3).

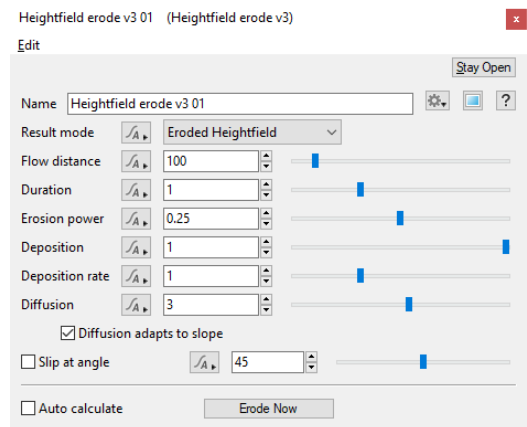
Existujú tiež zásuvné moduly pre Blender, Rain Paint (kapitola 2.4.1) a Erosion Add-On (kapitola 2.4.2), dopĺňajúce túto funkcionality. Oba plugíny sú platené a dostupné z obchodu [Blender Market](#).

2.1 Terragen

Terragen ¹ je pokročilý program pre vytváranie a modifikovanie terénu a taktiež umožňuje renderovať obrázky a videá. Pre tvorbu terénu je využívaný aj vo filmovom priemysle známymi filmovými štúdiami ².

Nevýhodou Terragenu je jeho vysoká cena, 300, alebo 600 amerických dolárov za licenciu Creative, alebo Professional. V ponuke je aj verzia zdarma, tá má avšak mnoho obmedzení a nemôže byť použitá na komerčné účely.

Súčastou výbavy tohoto produktu je aj operátor pre eróziu terénu ³. Jedná sa o eróziu vodou, ktorá sa dá prispôbiť pomocou niekoľkých parametrov: maximálna vzdialenosť toku (Flow distance), celková doba erózie (duration), sila erózie (erosion power), množstvo usadzovacieho sedimentu (deposition), rýchlosť usadzovania sedimentu (deposition rate) a drsnosť/tvrdosť terénu (diffusion).

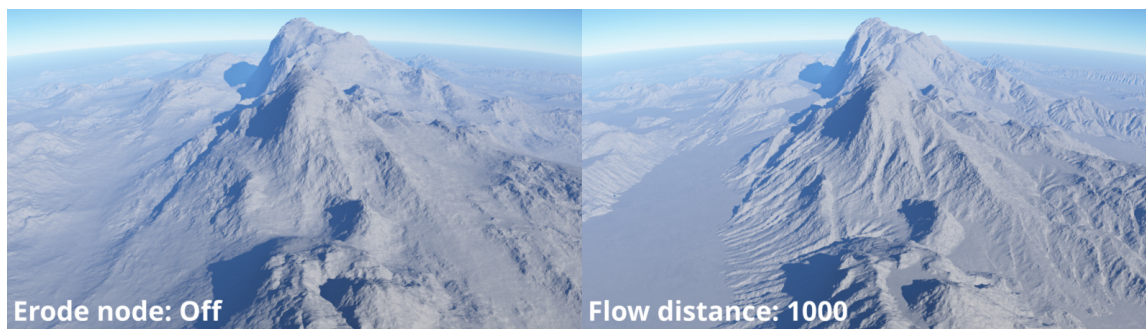


Obrázek 2.1: Nástroj na eróziu terénu programu Terragen. Prevzaté z webu planetside.co.uk.

¹Web nástroja Terragen: <https://planetside.co.uk/>

²Terragen vo filmovom priemysle: <https://planetside.co.uk/terrigen-in-film-and-tv/>

³Operátor erózie terénu v nástroji Terragen: [https://planetside.co.uk/wiki/index.php?title= Heightfield_Erode_v3](https://planetside.co.uk/wiki/index.php?title=Heightfield_Erode_v3)



Obrázek 2.2: Terén pred eróziou. Prevzaté zObrázek 2.3: Terén po erózii. Prevzaté z webu planetside.co.uk.

2.2 Unreal Engine

Unreal Engine ⁴ je herný engine od spoločnosti Epic a patrí medzi najpoužívanejšie ⁵. Používa sa na tvorbu rozsiahlych hier s modernou grafikou.

Medzi nástrojmi na úpravu terénu ponúka termálnu ⁶ a vodnú eróziu ⁷.

Obrázok 2.5 zobrazuje nastavenia dostupné v nástroji na termálnu eróziu. Základnými premennými sú sila erózie (Total Strength), maximálny rozdiel výšky (Threshold) a počet iterácií (Iterations). Navyše ponúka možnosť využiť šum pre prirodzenejší vzhľad erodovaného terénu.

Nastavenia vodnej erózie sú zobrazené na obrázku 2.6. Efekt erózie sa dá ovládať parametrami sila erózie (Total Strength), množstvo dažďa (Rain Ammount), maximálna kapacita sedimentu (Sediment Cap) a počet iterácií (Iterations). Ovládať sa dá aj rozmiestnenie dažďa, teda prínosu vody do simulácie, na rovnomerné, alebo náhodné.



**UNREAL
ENGINE**

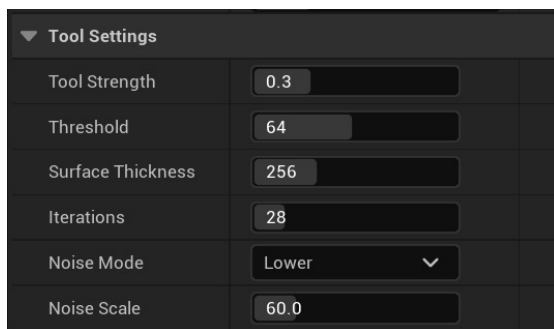
Obrázek 2.4: Unreal Engine logo. Prevzaté z unrealengine.com.

⁴Webová stránka herného engine Unreal Engine: <https://www.unrealengine.com/>

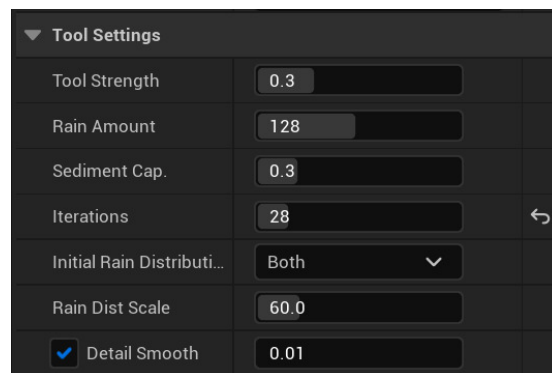
⁵Stack Overflow Survey: <https://insights.stackoverflow.com/survey/2021/>

⁶Nástroj termálnej erózie v hernom engine Unreal Engine: <https://docs.unrealengine.com/5.0/en-US/landscape-erosion-tool-in-unreal-engine/>

⁷Nástroj vodnej erózie v hernom engine Unreal Engine: <https://docs.unrealengine.com/5.0/en-US/landscape-hydro-erosion-tool-in-unreal-engine/>



Obrázek 2.5: Panel nástroja termálnej erózie v Unreal Engine. Prevzaté z unrealengine.com.



Obrázek 2.6: Panel nástroja vodnej erózie v Unreal Engine. Prevzaté z unrealengine.com.

2.3 Unity

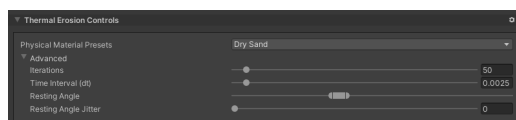
Unity ⁸ je najpoužívanejší herný engine na trhu ⁹ a najviac je využívaný strednými a menšími hernými štúdiami, v malých tímoch alebo sólo vývojároch.

Taktiež ako aj Unreal Engine, má vstavané nástroje na prácu s terénom. V ponuke má termálnu, hydraulickú a veternú eróziu terénu s najdetailnejšími nastaveniami v porovnaní s konkurenciou ¹⁰.

Panel nástroja pre termálnu eróziu, zobrazený na obrázku 2.8, je podobný tomu v Unreal Engine. Obsahuje nastavenia iterácii (Iterations) a maximálny uhol svahu, pri ktorom je terén stabilný a nezosúva sa (Resting Angle). V ponuke je niekoľko prednastavených profilov pomenovaných podľa materiálu ktorého erózia sa má simulovať. Tieto profily definujú hodnoty premenných v kategórii „pokročilé“ (Advanced).



Obrázek 2.7: Unity logo. Prevzaté z unity.com.



Obrázek 2.8: Panel nástroja termálnej erózie v Unity. Prevzaté z unity3d.com.

Obrázok 2.9 zobrazuje panel nástroja pre vodnú eróziu. Okrem základných nastavení, ktoré ponúkajú konkurenčné nástroje, ako počet iterácii (Iterations), množstvo zrážok (Precipitation rate), alebo maximálna kapacita erodovaného sedimentu (Sediment capacity), sprístupňuje Unity pokročilejšie nastavenia ovplyvňujúce vlastnosti transportu vody a sedimentu, alebo brehu rieky (Riverbank).

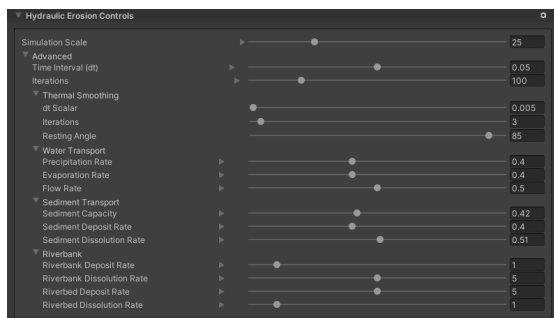
Unity ako jediný zo spomenutých nástrojov ponúka eróziu pôsobením vetra, zobrazený je na obrázku 2.10. Simulácia sa dá nastavovať do podrobná pomocou veľkému počtu premen-

⁸Webová stránka herného enginu Unity: <https://unity.com/>

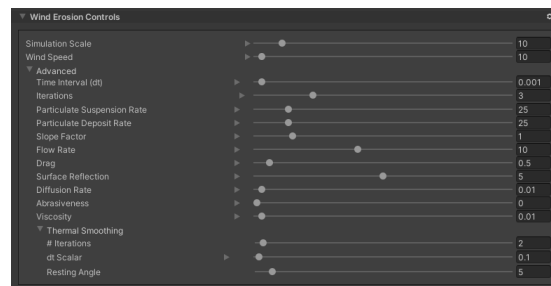
⁹Stack Overflow Survey: <https://insights.stackoverflow.com/survey/2021/>

¹⁰Nástroj pre eróziu v Unity: <https://docs.unity3d.com/Packages/com.unity.terrain-tools@4.0/manual/erosion.html>

ných. Najvplyvnejšie hodnoty sú škála simulácie (Simulation Scale), ktorá ovláda veľkosť črt ktoré erózia vytvára, a rýchlosť vetra (Wind Speed), ktorá ovplyvňuje silu vizuálneho efektu erózie.



Obrázek 2.9: Panel nástroja vodnej erózie v Unity. Prevzaté z unity3d.com.



Obrázek 2.10: Panel nástroja veternej erózie v Unity. Prevzaté z unity3d.com.

2.4 Blender

Blender [4] je bezplatný a open-source nástroj pre prácu s 3D grafikou. Používa sa na tvorbu 3D modelov, animácií, fyzikálnych simulácií, vizuálnych efektov a renderovania obrázkov, alebo videí. Funkcionalitou sa porovnáva s drahými komerčnými nástrojmi ako Autodesk Maya ¹¹, alebo Cinema 4D ¹².

Blender nedisponuje nástrojom pre eróziu terénu, avšak existujú zásuvné moduly ktoré túto funkcionality dopĺňajú.



Obrázek 2.11: Blender logo. Prevzaté z blender.org.

2.4.1 Rain Paint

Rain Paint je zásuvný modul pre Blender ktorý umožňuje simulovať vodnú eróziu terénu. Je dostupný z Blender Marketu za sumu \$20.

Na obrázku 2.12 je zobrazený panel s nastaveniami. Nakonfigurovať sa dá intenzita dažďa (Rain intensity), životnosť kvapky (Drop lifetime), krehkosť materiálu (Ground brittleness) a objem jednej kvapky (Drop volume) Zaujímavá je možnosť vymedzenia plochy na ktorej prší, pomocou skupiny bodov (Vertex Group) vytvorenej v Blenderi.

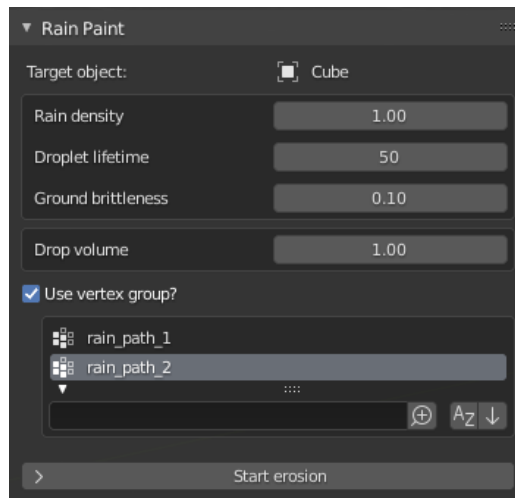
2.4.2 Erosion Add-On

Obrázok 2.13 zobrazuje panel zásuvného modulu Erosion Add-On doztupný z Blender Marketu za sumu \$49. Tento plugin je pokročilejší, pracuje s UV mapami a preto dokáže erodovať aj zložitejšie terény obsahujúce prevysy, alebo tunely.

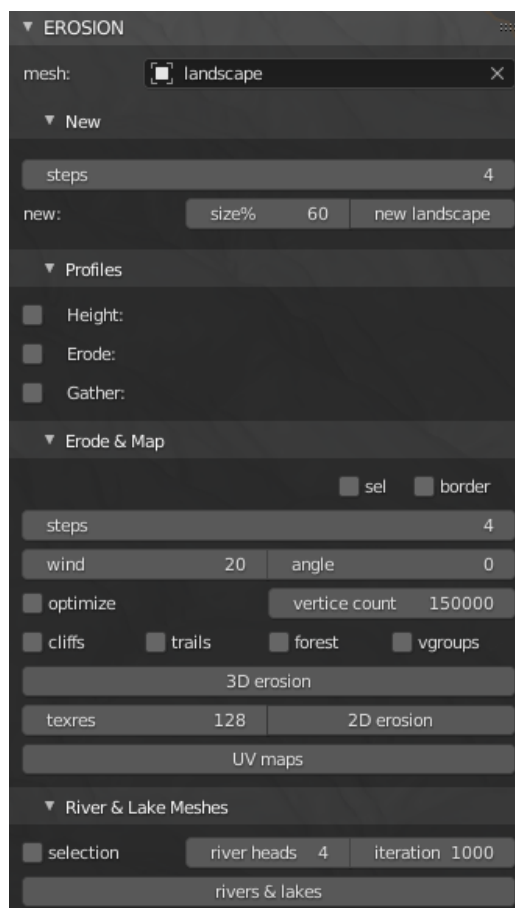
Tento nástroj ponúka širšiu škálu nastavení, ako je možné vidieť na obrázku nižšie.

¹¹Autodesk Maya: <https://www.autodesk.com/products/maya/>

¹²Cinema 4D: <https://www.maxon.net/en/cinema-4d/>



Obrázek 2.12: Panel zásuvného modulu Rain Paint.



Obrázek 2.13: Panel zásuvného panelu Erosion Add-On.

Kapitola 3

Metódy erózie

Erózia terénu je téma ktorou sa zaoberajú výskumníci z oblasti počítačovej grafiky viac než tri desaťročia. Už v roku 1989 bol uverejnený článok v časopise Computer Graphics [6] popisujúci 2 metódy erózie terénu, ktoré sa stali základom pre pokročilejšie metódy vyvinuté v nasledujúcich rokoch.

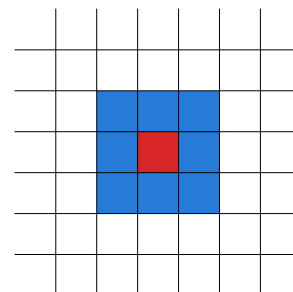
Prvou metódou je termálna erózia (kapitola 3.2), ktorá napodobňuje prirodzený rozpad hornín a následný zosuv pôdy. Tou druhou je vodná erózia (kapitola 3.3), simulujúca pôsobenie vody na terén, rozklad pôdy na sediment, ktorý je zadržaný vo vode, následný transport na nižšie miesta, kde sa sediment opäť usadí.

V tejto kapitole sú bližšie popísané metódy termálnej a vodnej erózie a ich možné implementácie.

3.1 Celulárny automat

Celulárny automat sa používa ako model vo viacerých metódach erózie. Pôvodný terén je reprezentovaný ako výšková mapa, teda 2D pole, ktorá obsahuje výšky jednotlivých bodov. Tieto body sú reprezentované ako bunky v celulárnom automate, rozložené v pravouhlej mriežke s rovnomernými vzdialenosťami. Každá bunka môže udzŕiavať niekoľko hodnôt, ktoré definujú jej stav.

Nižšie popísané algoritmy považujú za susedné bunky, bunky z Moorovho okolia, zobrazené na obrázku 3.1, kde je červenou farbou reprezentovaná aktuálna bunka a modrou jej okolie, v tomto prípade 8 susedných buniek. Výber iného okolia môže ovplyvniť výsledný efekt erózie.



Obrázek 3.1: Červenou farbou je zvýraznená aktuálna bunka a modrou jej susedné bunky z Moorovho okolia

3.2 Termálna erózia

Termálna erózia patrí medzi jednoduchšie metódy. Napodobňuje prirodzený rozpad hornín na menšie kusy, spôsobený zmenami teploty, a ich následné zosúvanie zo svahov pôsobením gravitácie. Efekt erózie je najvýraznejší v strmých svahoch a znižuje sa postupne s tým, ako sa znižuje uhol terénu až po hraničnú hodnotu, nazývanú talus. Terén s náklonom menším než je hodnota talusu sa nezosúva, je stabilný. Simulácia erózie tak môže

zkonvergovať do finálneho stavu, na ktorý už nebude mať žiadny vplyv bez zmeny tejto hraničnej hodnoty.

Ako model, sa pre tento algoritmus popísaný v článkoch The Synthesis and Rendering of Eroded Fractal Terrains [6] a Realtime Procedural Terrain Generation [7] používa celulárny automat (kapitola 3.1). Každá bunka obsahuje jednu hodnotu, výšku terénu h v danom bode.

Algoritmus sa vykonáva v iteráciách, v každej iterácii sa vyhodnotia podmienky pre každú bunku jednotlivo. Vypočíta sa rozdiel výšky medzi aktuálnou bunkou a bunkami v jej okolí. Ak rozdiel prekračuje konštantu talus T , časť pôdy je presunutá do okolia. Toto množstvo je udané percentuálnou čiastkou c z rozdielu týchto hodnôt. Podľa J. Olsena [7] je pre c vhodná hodnota 0.5. Vyššie hodnoty môžu spôsobiť osciláciu, pretože zmeny sú aplikované po jednej iterácii, naopak, nižšie hodnoty spôsobia pomalšie približovanie sa k maximálnemu uhlu svahu T a zvýši sa počet iterácií potrebných pre dosiahnutie rovnakého efektu erózie.

$$h_i = \begin{cases} d_i > T : & h_i + c \cdot (d_i - T) \\ d_i \leq T : & h_i \end{cases} \quad (3.1)$$

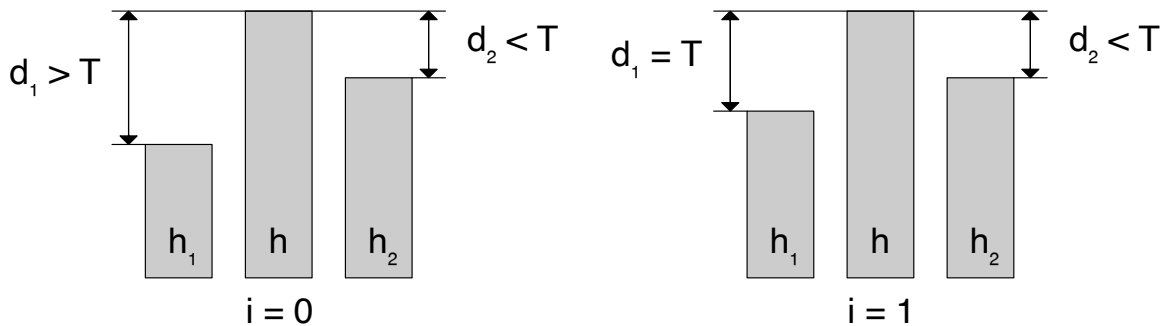
Vo vyššie uvedenej rovnici reprezentuje premenná h_i výšku i -tej bunky, kde i je index bunky z okolia, ako je znázornené na obrázku 3.2. Delta d_i je rozdiel výšky medzi aktuálnou bunkou a i -tou bunkou, matematicky vyjadrené ako $d_i = h - h_i$.

Pre použitie v zmieňovanom celulárnom automate, je potrebné rovnomerne distribuovať zosúvanú pôdu do okolia, pomocou nasledovnej rovnice:

$$h_i = h_i + c \cdot (d_{max} - T) \cdot \frac{d_i}{d_{total}} \quad (3.2)$$

kde d_{max} je maximum z hodnôt d_i a d_{total} je suma d_i , ktoré sú väčšie než T .

Na obrázku 3.3 je zobrazený príklad procesu termálnej erózie v nulte a prvej iterácii.



Obrázek 3.3: Príklad iterácie 0 a 1 termálnej erózie. Rozdiel výšok d_1 je väčší než talus a preto je časť terénu z bunky h zosunutá do bunky h_1 . V iterácii 1 sú oba rozdiely výšok menšie alebo rovné talusu, ďalšia erózia už nenastáva. Inšpirované z článku Realtime Procedural Terrain Generation [7].

h_1	h_2	h_3
h_4	h	h_5
h_6	h_7	h_8

Obrázek 3.2: Indexovanie susedných buniek. Inšpirované z článku Realtime Procedural Terrain Generation [7].

3.3 Vodná erózia

Vodná (hydraulická) erózia simuluje vplyv vody na terén a je zložitejšia na implementáciu aj výpočet v porovnaní s termálnou eróziou. Závisí na prínose vody do systému. Môže sa jednať o rovnomerné alebo náhodné rozloženie po celej ploche, napodobňujúce dážď, alebo lokálne na vyhradenej oblasti terénu, ako prameň alebo iný zdroj vody. Voda následne rozpustí časť materiálu (sediment) a transportuje ho na nižšie položené miesta, kde sa opäť usádza, spolu s tým ako sa voda odparuje a jej schopnosť zadržiavať sediment znižuje. Tento proces zanecháva v erodovanom teréne prirodzené prvky ako sú žľaby a údola.

Vodná erózia sa dá implementovať pomocou celulárneho automatu (kapitola 3.3.1), alebo simulovaním častíc (kapitola 3.3.2).

3.3.1 Implementácia pomocou celulárneho automatu

Implementácia vodnej erózie pomocou celulárneho automatu (kapitola 3.1) je jeden z možných prístupov, ktorým sa zaoberajú J. Olsen v článku Realtime Procedural Terrain Generation [7] a Bedřich Beneš et al. v článkoch Visual Simulation of Hydraulic Erosion [1] a Interactive Terrain Modeling Using Hydraulic Erosion [9]. Algoritmus je možné bez veľkých zmien paralelizovať a akcelerovať na GPU ako demonštrovali Balázs Jákó v práci Fast Hydraulic and Thermal Erosion on the GPU [8] a Xing Mei et al. v článku Fast Hydraulic Erosion Simulation and Visualization on GPU [5].

Nevýhodou tejto metódy je, že sa v každej iterácii počíta nový stav všetkých buniek, aj v prípade že je voda prítomná iba na časti terénu.

Každá bunka si uchováva jej stav v 3 premenných: výška bodu h , množstvo vody w a množstvo sedimentu m . Jadro algoritmu sa skladá z nasledujúcich krokov ktoré sa vykonávajú každú iteráciu

1. Prísun vody do systému
2. Rozpustenie časti terénu na sediment
3. Distribúcia vody a sedimentu
4. Vyparenie vody a usadzovanie sedimentu

V prvom kroku je do systému pridaná voda. Môže byť distribuovaná rovnomerne vo všetkých bunkách alebo náhodne, prípadne môže byť jej prísun obmedzený na časti terénu. Prísun vody pre bunku i bude nasledovný:

$$w_i = w_i + K_r \quad (3.3)$$

kde w_i je množstvo vody v i -tej bunke a K_r je konštanta udávajúca množstvo vody, ktoré pribudne do každej bunky v jednej iterácii. Alternatívne môže byť táto konštanta nahradená funkciou, napríklad funkciou šumu, ktorá určí množstvo pridanej vody pre danú bunku.

V druhom kroku je simulované rozpustenie materiálu vodou, na sediment.

$$\begin{aligned} h_i &= h_i - K_s \cdot w_i \\ m_i &= m_i + K_s \cdot w_i \end{aligned} \quad (3.4)$$

Pomer množstva sedimentu, ktorý voda dokáže rozpustiť je daný konštantou K_s

V treťom kroku sú voda a v nej rozpustený sediment transportované do nižšie položených buniek. Celková výška bunky, označená premennou a je súčet výšky a vody v bunke:

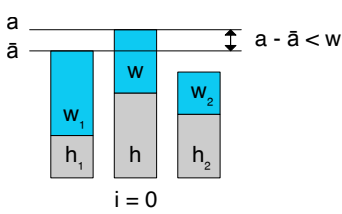
$$a_i = h_i + w_i \quad (3.5)$$

Na distribúciu sa podielajú iba susedné bunky, ktoré sú nižšie než aktuálna bunka. Množstvo vody presunutej do susednej bunky je vypočítané nasledovne:

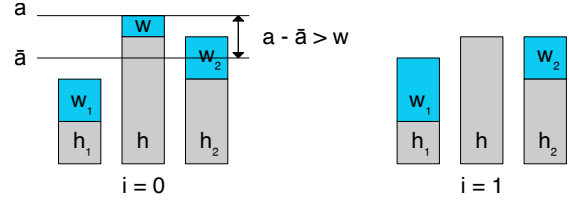
$$\Delta w_i = \min(w, \Delta a) \cdot \frac{d_i}{d_{total}} \quad (3.6)$$

kde $\Delta a = a - \bar{a}$ je celková výška aktuálnej bunky mínus aritmetický priemer celkovej výšky buniek zahrnutých v distribúciu, vrátane aktuálnej bunky, d_i je rozdiel výšky aktuálnej bunky a i -tej bunky $d_i = a - a_i$ a d_{total} je súčet všetkých kladných hodnôt d_i .

V prípade že je množstvo vody menšie než rozdiel výšky, je jej celý objem presunutý do susednej bunky. V opačnom prípade je voda rozdelená tak, že sa distribúciou vyrovná celková výška susedných buniek. Oba príklady sú zobrazené na obrázkoch 3.4 a 3.5.



Obrázek 3.4: Prípád 1: aktuálna bunka obsahuje viac vody, než je potrebné pre vyrovnanie celkovej výšky susedných buniek. Inšpirované z [7].



Obrázek 3.5: Prípád 2: aktuálna baunka obsahuje menej vody, než je potrebné pre vyrovnanie celkovej výšky susedných buniek a distribuje celý svoj objem vody. Inšpirované z [7].

Spolu s vodou je rovnomerným množstvom distribuovaný aj sediment:

$$\Delta m = m \cdot \frac{\Delta w_i}{w} \quad (3.7)$$

V poslednom kroku je časť vody vyparená. Podiel vyparenej vody je daný konštantou K_e , množstvo vody po vyparení je dané rovnicou:

$$w = w \cdot (1 - K_e) \quad (3.8)$$

Voda môže obsahovať maximálne určité množstvo sedimentu. Toto množstvo je dané koeficientom, konštantou K_c :

$$m_{max} = K_c \cdot w \quad (3.9)$$

V prípade že celková koncentrácia vo vode presiahne daný pomer, musí sa prebytočný sediment usadiť.

$$\Delta m = \max(0, m - m_{max})m = m - \Delta m h = h + \Delta m \quad (3.10)$$

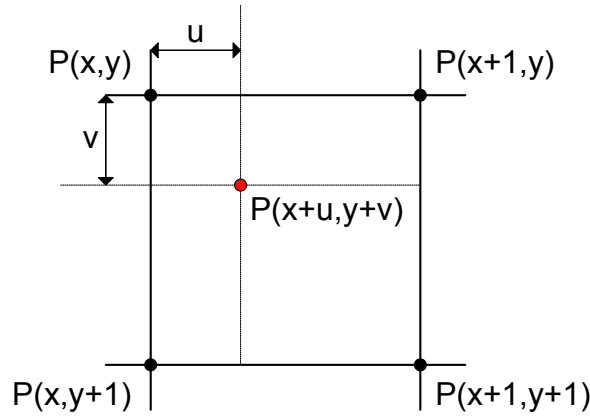
Po dokončení všetkých iterácií, je potrebné usadiť zvyšný sediment $h_i = h_i + m_i$, aby nedošlo k strate hmoty z terénu.

3.3.2 Implementácia pomocou simulácie častíc

Táto metóda [2] má iný prístup k simulácii vody, tým že je voda reprezentovaná jednotlivými kvapkami, ktoré sa pohybujú po teréne. Terén je reprezentovaný tak ako v predchádzajúcich implementáciach, pravouhlou mriežkou s rovnako vzdialenými bodmi. Výpočetne je menej náročná než implementácia pomocou celulárneho automatu, pretože simuluje iba pohyb kvapky a jej efekty na terén v danom mieste, ale jej nevýhodou je, že nie je veľmi presná, pretože simuluje jednotlivé kvapky samostatne.

Namiesto počtu iterácií sa používa počet kvapiek ktoré sa majú odsimulovať. Životnosť kvapky môže byť ovplyvnená vyparovaním, alebo limitom maximálnej cesty, ktorú môže prejsť. Zaujímavou vlastnosťou tejto implementácie je, že sa simulované kvapky pohybujú iba v 2D priestore a pohybujú sa vždy rovnakú vzdialenosť a preto nie je simulovaný čas pre každý krok rovnako dlhý.

Každá kvapka obsahuje jej pozíciu pos reprezentovanú 2D vektorom a smer pohybu dir reprezentovaný ako jednotkový 2D vektor. Taktiež si uchováva informácie o množstve vody $water$ a sedimentu $sediment$.



Obrázek 3.6: Kvapka, zvýraznená červenou farbou, medzi 4 susednými bodmi. Vyznačené hodnoty sú použité pri bilineárnej interpolácii.

Pre posun kvapky je potrebné vypočítať gradient v jej mieste pos_{old} bilineárnou interpoláciou gradientov 4 susedných bodov. Napríklad pre pozíciu $pos_{old} = (x + u, z + v)$, kde $u, v \in < 0, 1 >$ je vzdialenosť kvapky od bodu (x, y) . Gradienty susedných bodov $P_{x,y}$, $P_{x+1,y}$, $P_{x,y+1}$ a $P_{x+1,y+1}$, zvýraznených na obrázku 3.6, sú:

$$\begin{aligned}
 g(P_{x,y}) &= \begin{pmatrix} P_{x+1,y} - P_{x,y} \\ P_{x,y+1} - P_{x,y} \end{pmatrix} \\
 g(P_{x+1,y}) &= \begin{pmatrix} P_{x+1,y} - P_{x,y} \\ P_{x+1,y+1} - P_{x+1,y} \end{pmatrix} \\
 g(P_{x,y+1}) &= \begin{pmatrix} P_{x+1,y+1} - P_{x,y+1} \\ P_{x,y+1} - P_{x,y} \end{pmatrix} \\
 g(P_{x+1,y+1}) &= \begin{pmatrix} P_{x+1,y+1} - P_{x,y+1} \\ P_{x+1,y+1} - P_{x+1,y} \end{pmatrix}.
 \end{aligned} \tag{3.11}$$

Pomocou bilineárnej interpolácie je možné definovať gradient g v bode pos_{old} :

$$g(pos_{old}) = \begin{pmatrix} (P_{x+1,y} - P_{x,y}) \cdot (1 - v) + (P_{x+1,y+1} - P_{x,y+1}) \cdot v \\ (P_{x,y+1} - P_{x,y}) \cdot (1 - u) + (P_{x+1,y+1} - P_{x+1,y}) \cdot u \end{pmatrix}. \quad (3.12)$$

Tento 2D vektor je použitý pre určenie nového smeru dir_{new} . Výsledná hodnota je mixom hodnôt g a dir_{old} , určeným hodnotou parametru $p_{inertia} \in \langle 0, 1 \rangle$:

$$dir_{new} = dir_{old} \cdot p_{inertia} - g \cdot (1 - p_{inertia}) \quad (3.13)$$

kde hodnota 0 znamená že starý smer dir_{old} nie je braný do úvahy a výsledná hodnota je rovná negácii gradientu g . V opačnom prípade, keď je hodnota $p_{inertia}$ rovná 0, je zanedbaná hodnota gradientu g a smer zostáva nezmenený. Následne je vektor normalizovaný.

Nová pozícia kvapky pos_{new} je vypočítaná nasledovne:

$$pos_{new} = pos_{old} + dir_{new} \quad (3.14)$$

Keďže je vektor smeru dir_{new} normalizovaný, nová pozícia je vždy vzdialená jednu jednotku, bez ohľadu na rýchlosť kvapky vel a preto jeden krok simulácie nereprezentuje rovnaký časový úsek.

Vzdialenosť jednej jednotky je zvolená preto, že je to najväčšia vzdialenosť pri ktorej kvapka nedokáže preskočiť jeden, alebo viac bodov a vďaka tomu nedochádza k nekonzistentným stavom. Napríklad by kvapka mohla preskočením niekoľkých bodov neerodovať terén, keď mala a zanechala by v ceste kopec, alebo by neusadila sediment keď mala a nechala by v jej ceste jamu. V prípade výberu menšej vzdialenosti, by bolo potrebných viac krokov simulácie pre dosiahnutie podobného efektu erózie, čo je nežiadané.

Rozdiel výšky h_{dif} medzi predchádzajúcou polohou pos_{old} a aktuálnou pos_{new} je vypočítaný rovnicou

$$h_{dif} = h_{new} - h_{old} \quad (3.15)$$

kde výšky h_{new} a h_{old} sú taktiež interpolované z výšok okolitých bodov, poprípade h_{old} môže byť známa z predchádzajúceho kroku simulácie. Podľa tohoto rozdielu je možné určiť či kvapka cestovala hore, alebo dole kopcom.

V prípade že je hodnota h_{dif} kladná, nová poloha kvapky sa nachádza na vyššie položenom mieste a časť sedimentu sa usadí v starej polohe pos_{old} . Nastatť môže jeden z 2 scenárov, buď má kvapka dostatok sedimentu na vyplnenie celej jamy, alebo nemá a usadí všetok svoj sediment v snahe vyplniť jamu.

V prípade že je hodnota záporná, kvapka sa presunula na nižšie položené miesto a kvapka eroduje terén z predošlej lokality pos_{old} .

Vo výnimočných prípadoch môže byť hodnota rovná 0, kedy sa kvapka pohybuje po rovnom teréne a kapacita sedimentu konverguje k 0. Niekedy je žiadané erodovať aj rovný terén z estetických dôvodov, preto je do nasledujúcej rovnice pridaná konštanta $p_{minSlope}$, ktorá reprezentuje minimálny rozdiel výšky pri jednom kroku kvapky. Maximálna kapacita sedimentu sa vypočíta rovnicou:

$$c = \max(-h_{dif}, p_{minSlope}) \cdot vel \cdot water \cdot p_{capacity} \quad (3.16)$$

V prípade že kvapka obsahuje viac sedimentu než je jej maximálna kapacita, časť sedimentu sa usadí v predchádzajúcej lokalite pos_{old} .

V opačnom prípade, keď kvapka obsahuje menej sedimentu než jej maximálna kapacita dovoľuje, eroduje časť terénu z predchádzajúcej lokality, udanú percentuálne konštantou $p_{erosion}$. Množstvo sedimentu, ktorý sa usadí v pos_{old} je vypočítaný nasledujúcou rovnicou:

$$(sediment - c) \cdot p_{deposition} \quad (3.17)$$

kde konštanta $p_{deposition}$ vyjadruje rýchlosť usádzania.

Dôležité je zaistiť aby kvapka neerodovala viac sedimentu než je rozdiel výšok h_{dif} , aby nevznikali jamy v teréne. Nasledujúca rovnica to berie v úvahu a vyjadruje konečné množstvo erodovaného terénu z pos_{old} :

$$\min((c - sediment) \cdot p_{erosion}, -h_{dif}) \quad (3.18)$$

Následovne je vypočítaná nová rýchlosť kvapky a časť jej vody sa vyparí:

$$\begin{aligned} vel_{new} &= \sqrt{vel_{old}^2 + h_{dif} \cdot p_{gravity}} \\ water_{new} &= water_{old} \cdot (1 - p_{evaporation}) \end{aligned} \quad (3.19)$$

kde parameter $p_{gravity}$ reprezentuje gravitačnú silu.

Proces erózie je ovplyňovaný konštantou p_{radius} , ktorá udáva polomer vplyvu kvapky. Body terénu vrámci tejto vzdialenosti od kvapky sú zahrnuté v erózii, kde každý bod P_i prispieva váhou w_i vypočítanou pomocou rovnice:

$$w_i = \frac{\max(0, p_{radius} - (|P_i - pos|))}{\sum_{k=0}^n \max(0, p_{radius} - (|P_k - pos|))} \quad (3.20)$$

kde n je počet bodov v rozsahu polomeru p_{radius} od kvapky.

Pri erózii sa taktiež berie ohľad na tvrdosť materiálu, udávanú funkciou, ktorej vstupom je poloha bodu a jej výstupom je hodnota v intervale $< 0, 1 >$. Hodnota 0 znamená že sa žiaden materiál neeroduje, naopak hodnota 1 znamená že sa eroduje celá čiastka určená váhou w_i :

$$w'_i = w_i \cdot f(P_{ix}, P_{iy}, P_{iz}) \quad (3.21)$$

Pomocou tejto funkcie je možné simulovať rôzne zloženie terénu a neerodovaním niektorých častí dosiahnuť tvary ako sú skaly, alebo kamene.

Kapitola 4

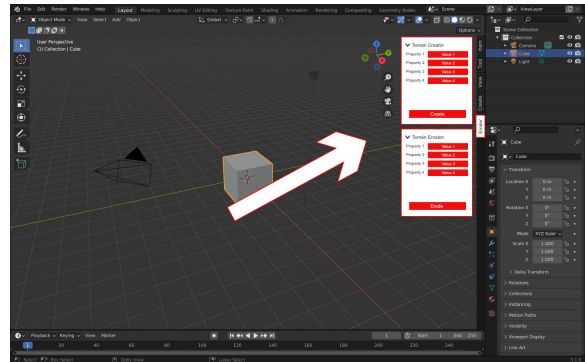
Návrh zásuvného modulu pre eróziu

Cieľom je vytvoriť zásuvný modul pre Blender ktorý umožní eróziu objektu - terénu. Tento plugin by mal byť bezplatný, verejne dostupný, jednoducho inštalovateľný a užívateľsky prívetivý.

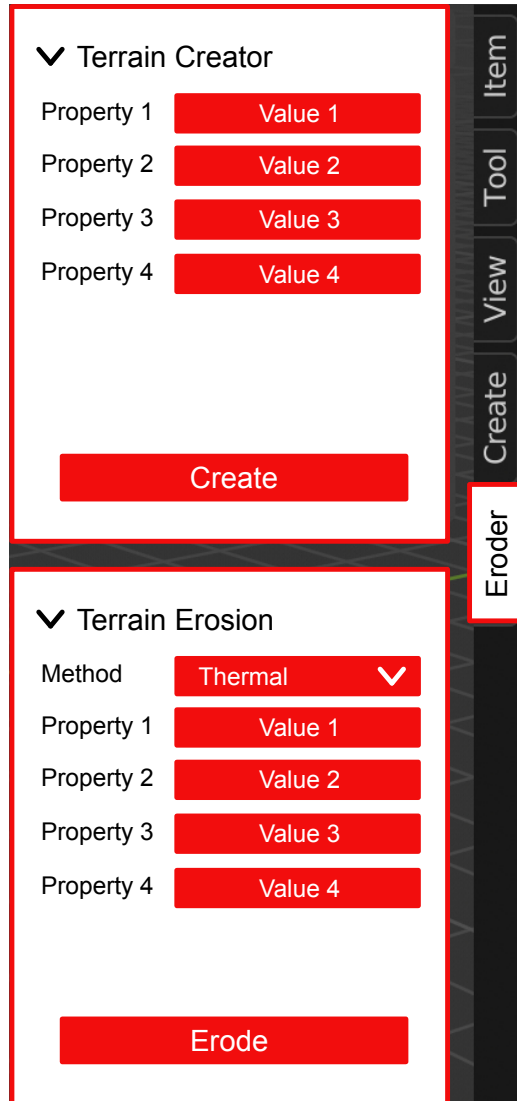
Užívateľským rozhraním je panel umiestnený v pravej časti aplikačného okna, ako je vidieť na obrázku 4.1 a jeho detail je zobrazený na obrázku 4.2.

Podpanel pre eróziu terénu sa nachádza v dolnej časti. Drop-down prvok má v ponuke 3 typy erózie, termálnu, vodnú implementovanú pomocou celulárneho automatu a vodnú implementovnú pomocou simulácie častíc. Každá metóda obsahuje relevantné premenné, jednoducho nastavovateľné vstavanými prvkami užívateľského rozhrania. Úplne dole sa nachádza tlačidlo pre spustenie procesu erózie, ktoré sa po stlačení zmení na indikátor stavu prebiehajúcej simulácie.

Druhý podpanel, nachádzajúci sa v hornej časti detailu návrhu 4.2, plní funkcionality vytvárania terénu pomocou šumov z matematickej knižnice `mathutils`. Zásuvný modul tak môže byť používaný v celom procese vytvárania realistického terénu, od procedurálneho generovania terénu, po jeho eróziu vybranou metódou. V ponuke je niekoľko typov šumu, a ich špecifické nastavenia v podobe premenných, ktoré produkujú rôznorodý terén. Nechýba ani možnosť konfigurovať veľkosť a rozlíšenie generovaného terénu. Ako posledné je v paneli tlačidlo, ktoré spustí akciu pre vytvorenie terénu s danými parametrami a umiestni ho do aktuálnej Blender scény.



Obrázek 4.1: Mockup - umiestnenie zásuvného modulu.



Obrázek 4.2: Mockup - detail.

Kapitola 5

Implementácia

V tejto kapitole sú popísané využité technológie a nástroje použité pri tvorbe tohoto projektu a postup pri implementácii. Zahŕňa detaily o vytvorení zásuvného modulu (kapitola 5.2) pre Blender, jeho užívateľského rozhrania, definovania a využitia (vlastných) premenných v Blenderi (kapitola 5.3) a implementácii termálnej (kapitola 5.5) a vodnej erózie (kapitoly 5.6 a 5.7)

5.1 Použité technológie a nástroje

Vývoj prebiehal na operačných systémoch Windows a macOS, pre verzovanie kódu bol použitý verzovací systém Git ¹, spolu s cloudovou službou GitHub ² pre zálohovanie a synchronizáciu medzi zariadeniami. Kód bol editovaný pomocou editora Visual Studio Code (kapitola 5.1.8) a nápovedu pre automatické dopĺňanie kódu poskytol Python balíček `fake-bpy-module` (kapitola 5.1.9).

5.1.1 Blender

Blender je nástroj pre prácu s 3D grafikou, bližšie už bol popísaný v kapitole 2.4. Pri vývoji bola použitá verzia 3.1.0, dostupná na stiahnutie z webu³.

5.1.2 Python

Python ⁴ je interpretovaný a dynamicky typovaný (skriptovací) programovací jazyk, ktorý sa používa na tvorbu zásuvných modulov pre Blender. Tento jazyk je interpretovaný a dynamicky typovaný. Patrí medzi medzi najpoužívanejšie technológie a je medzi programátormi veľmi obľúbený ⁵.



Obrázek 5.1: Python logo. Prevzaté z python.org.

¹Verzovací systém Git: <https://git-scm.com/>

²Cloudová služba GitHub: <https://github.com/>

³Blender na stiahnutie: <https://www.blender.org/download/>

⁴Webová stránka jazyka Python: <https://www.python.org/>

⁵Stack Overflow Survey: <https://insights.stackoverflow.com/survey/2021/>

5.1.3 Blender API

Blender API [3] umožňuje interagovať s Blenderom pomocou Python skriptov. Blender obsahuje Python interpret ktorý je spustený pri štarte Blenderu a počas jeho behu interpretuje skripty vytvorené užívateľmi, zásuvné moduly, alebo interné nástroje. Tieto skripty môžu vykresľovať prvky užívateľského prostredia, alebo modifikovať vnútorný stav aplikácie.

Blender 3.1.0 použitý v tejto práci obsahuje Python interpret vo verzii 3.10.2. Tento interpret obsahuje navyše moduly z Blender API, ako napríklad `bpy`, `bmesh`, alebo `mathutils`.

5.1.4 Modul `bpy`

Modul `bpy` je hlavný modul Blender API obsahujúci typy a funkcie pre prácu s užívateľským prostredím, alebo prístup k stavu aplikácie, 3D scéne a jej objektom.

5.1.5 Modul `bmesh`

Modul `bmesh` umožňuje prácu s objektami typu `MESH`, teda objekty reprezentované bodmi v priestore. Reprezentácia a práca s týmto typom objektov je bližšie popísaná v kapitole 5.4.1.

5.1.6 Modul `mathutils`

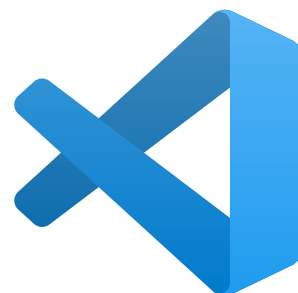
Modul `mathutils` obsahuje funkcionality pre matematické výpočty, napríklad modul `mathutils.noise` obsahuje funkcie pracujúce so šumom, ktoré sú v tejto práci využité na vytváranie terénu (kapitola 5.4).

5.1.7 Modul `numpy`

Modul `numpy` nie je súčasťou Blender API, ale je súčasťou Python distribúcie Blenderu. Tento modul umožňuje prácu s n-rozmernými polami a rýchlymi operáciami nad nimi, pomocou vektorizácie a iných techník.

5.1.8 Visual Studio Code

Visual Studio Code, v skratke VS Code, je textový editor od spoločnosti Microsoft ⁶. Funkcionalitou je veľmi podobný integrovaným vývojovým prostrediam vďaka jeho modularite pomocou pluginov, ktorá umožňuje nakonfigurovať prostredie pre prácu s ľubovoľným jazykom, alebo technológiou. Medzi programátormi sa stal veľmi obľúbený, čo ukazuje aj prieskum na stránke Stack Overflow ⁷, kde sa umiestnil na prvom mieste medzi najpoužívanejšími vývojovými prostrediami a textovými editormi.



5.1.9 Python balíček `fake-bpy-module`

Balíček `fake-bpy-module`⁸ sprístupňuje automatické dopĺňanie kódu, ako je vidieť na obrázku 5.3.

Obrázek 5.2: Visual Studio Code logo. Prevzaté z visualstudio.com.

⁶Textový editor Visual Studio Code: <https://code.visualstudio.com/>

⁷Stack Overflow Survey: <https://insights.stackoverflow.com/survey/2021/>

⁸Python balíček `fake-bpy-module`: <https://github.com/nutti/fake-bpy-module>



Obrázek 5.3: Automatické dopĺňanie kódu vo VS Code pomocou python balíčka `fake-bpy-module`.

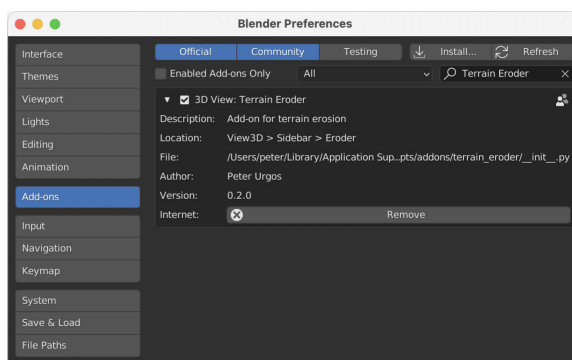
5.2 Zásuvný modul pre Blender

Vstupným bodom zásuvného modulu je súbor (Python skript), ktorý obsahuje slovník `bl_info` s metadátaami daného modulu, ako napríklad názov (`name`), autor (`author`, verzia modulu (`version`), minimálna podporovaná verzia Blenderu (`blender`), alebo popis (`description`). Výpis 5.1 zobrazuje obsah slovníka `bl_info` definovaného v tomto projekte, bez definovania tohoto slovníka by skript nebol rozpoznávaný ako zásuvný modul.

```
bl_info = {
    "name": "Terrain Eroder",
    "author": "Peter Urgos",
    "version": (0, 2, 0),
    "blender": (2, 93, 0),
    "location": "View3D > Sidebar > Eroder",
    "description": "Add-on for terrain erosion",
    "category": "3D View"
}
```

Výpis 5.1: Ukážka slovníka `bl_info`.

Proces inštalácie je rozdielny pre samostatný skript a pre projekt obsahujúci viacero skriptov v Python module, so vstupným skriptom nazvaným, podľa konvencie, `__init__.py`.



Obrázek 5.4: Nainštalovaný plugin zobrazený v nastaveniach Blenderu.

V prvom prípade stačí na inštaláciu súbor skriptu samotný, ale v tom druhom je potrebné celý modul zabalit do `.zip` archívu. Následne treba v inštaláčnom okne

(Edit->Preferences...->Add-ons->Install...) vybrať daný skript, alebo .zip archív, kliknúť na tlačidlo **Install Add-on** a zásuvný modul by sa mal zobrazíť v zozname nainštalovaných. V prípade že nie je plugin povolený, treba zaškrtnúť tlačidlo vedľa názvu modulu (obrázok 5.4).

Pri vývoji je nepraktické manuálne preinštalovávať modul po každej zmene, preto bola využitá funkcia symbolického linku z adresára, kde sa inštalujú zásuvné moduly, na adresár projektu, obsahujúci aktuálny kód.

5.3 Uživatelské rozhranie

Ako vhodný umiestnenie pre tento zásuvný modul bol vybraný bočný panel, ktorý je umiestnený v pravej strane okna (obrázok 5.5). Zobrazovať a schovávať sa dá klávesou N. Bočný panel sa skladá z viacerých panelov, ktoré sú zaradené do kategórii zobrazených v podobe záložiek po pravej strane. Vždy je zobrazená maximálne jedna kategória panelov a panely je možné vrámci nej presúvať a schovávať/rozbalovať podľa potreby.

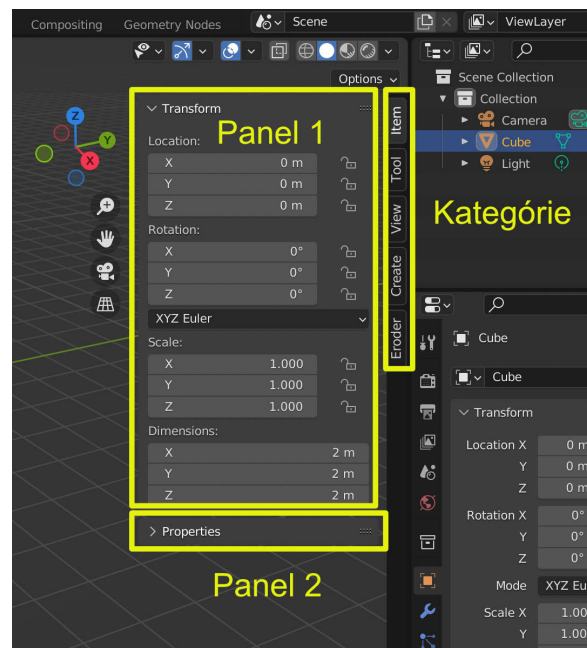
V Blender API je takýto panel reprezentovaný triedou `bpy.types.Panel`. Pre vytvorenie nového panelu je potrebné definovať novú triedu, ktorá z nej dedí, napríklad v tomto projekte je definovaný panel pre eróziu nasledovne: `class ErosionPanel(bpy.types.Panel)`

Následne je potrebné definovať niekoľko premenných s predponou `bl_`, ktoré kategorizujú daný panel. `bl_label` je názov panelu, `bl_idname` je ID panelu, konvencia je začať názov textom `OBJECT_PT_`, `bl_category` je kategória v ktorej sa má panel nachádzať, `bl_space_type` je priestor kde sa bude panel používať a `bl_region_type` je región kde sa bude panel používať. Panely tohoto zásuvného modulu sú umiestnené v priestore 3D viewportu a regióne UI.

Panel erózie ešte využíva premennú `bl_context`, ktorou obmedzuje viditeľnosť panelu iba na mód objektu, takže v móde editovania alebo iných nie je zobrazený.

Druhý panel, určený na vytváranie terénu definuje navyše premennú `bl_options = {"DEFAULT_CLOSED"}` a je nastavená na možnosť aby bol panel v pôvodnom stave schovaný.

```
class ErosionPanel(bpy.types.Panel):
    bl_label = "Terrain Erosion"
    bl_idname = "OBJECT_PT_terrain_erosion"
    bl_space_type = "VIEW_3D"
    bl_region_type = "UI"
    bl_context = "objectmode"
```



Obrázok 5.5: Bočný panel v Blenderi s vyznačenými kategóriami a panelmi. Panel 2 je schovaný.

```
bl_category = "Eroder"

def draw(self, context: bpy.types.Context):
    ...
```

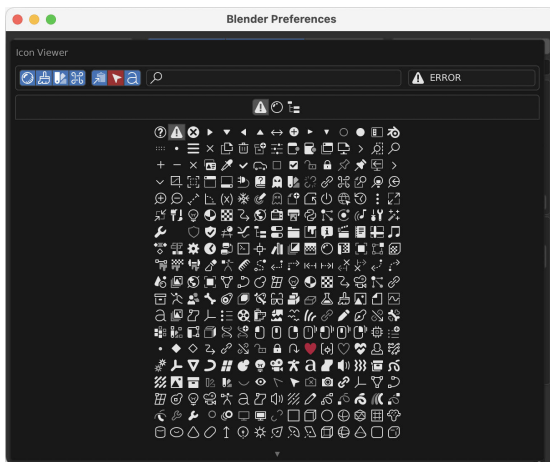
Výpis 5.2: Ukážka definície panela `bl_info`.

Ako je vidieť na výpise 5.2, v triede panelu je definovaná metóda `draw`. Táto funkcia je volaná opakovane a jej účelom je vytvárať prvky užívateľského rozhrania a umiestňovať ich v paneli. UI prvky sú dostupné cez objekt menom `layout` definovaný v triede panelu. Tento UI kontajner napríklad poskytuje metódu pre vytvorenie riadku `layout.row()`, do ktorého sa dajú vkladať ďalšie prvky (je taktiež UI kontajner) ako text, alebo premenné (properties). Kód vo výpise 5.3 ukazuje ako sa dá vytvoriť riadok a do neho vložiť text. Textové pole podporuje systémové ikony, ktoré sú definované v Blenderi a k ich zoznamu sa dá dostať pomocou zásuvného modulu `Icon Viewer` (obrázok 5.6).

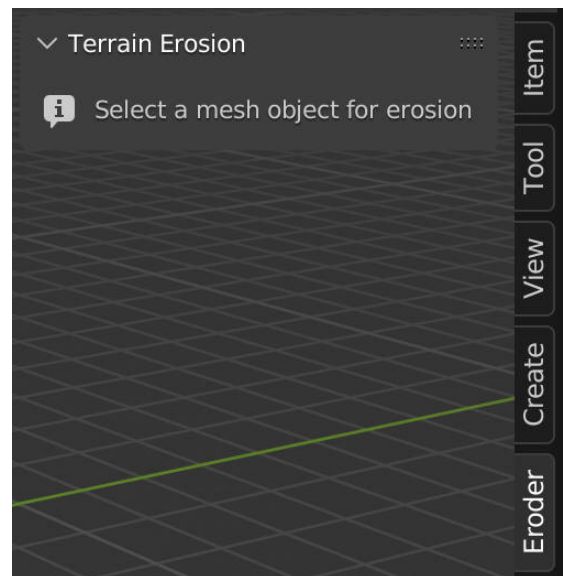
```
row = self.layout.row()
row.label(text="Select a mesh object for erosion", icon="INFO")
```

Výpis 5.3: Kód pre vytvorenie riadku s textom a ikonou v UI.

Napriek tomu že všetko potrebné definované, panel sa po spustení Blenderu nikde nezobrazí. Aby Blender definovaný panel `ErosionPanel` rozpoznal, je potrebné jeho triedu registrovať metódou `bpy.utils.register_class(ErosionPanel)`. Túto akciu treba vykonať pre každú novo-definovanú triedu, ktorá dedí funkcionality z Blender API. Po spustení by sa mal zobrazíť panel v definovanej kategórii (obrázok 5.7).



Obrázok 5.6: Zásuvný modul `Icon Viewer` zobrazujúci zoznam systémových ikon dostupných pre použitie v UI.



Obrázok 5.7: Ukážka panelu `Terrain Erosion` s textom a ikonou. Panel patrí do ním definovanej kategórie `Eroder`

5.3.1 Objekty typu `property`

Blender umožňuje zobrazovať a modifikovať stav špeciálnych objektov, zvaných `properties`. `Properties` reprezentujú bežné typy, ako normálne premenné používané v Pythone.

V Blenderi sú reprezentované tieto objekty adekvátnymi UI prvkami, ako napríklad slider pre číselné hodnoty, textové pole pre reťazce znakov a zaklikávacie políčko (checkbox) pre booleovské hodnoty.

V tomto projekte boli využité objekty property pre ukladanie parametrov erózie. Tieto premenné sú umiestnené v triede `ErosionProperties` ktorá dedí z triedy `bpy.types.PropertyGroup`. Trieda `PropertyGroup` slúži ako kontajner pre objekty typu `property`. Na výpise 5.4 je zobrazená deklarácia objektu `th_max_slope` typu `FloatProperty` v tejto triede.

```
class ErosionProperties(bpy.types.PropertyGroup):
    th_max_slope: FloatProperty(
        name="Max Slope",
        default=30,
        min=0,
        max=90,
        description="Maximum Slope Angle after which soil slides down"
    )

bpy.types.Scene.erosion_properties = PointerProperty(
    type=ErosionProperties,
    name="Erosion Properties",
    description="Properties of erosion")
```

Výpis 5.4: Deklarácia triedy `ErosionProperties` a jej atribútu typu `FloatProperty`. Následná inicializácia a uloženie do novej premennej v scéne Blenderu.

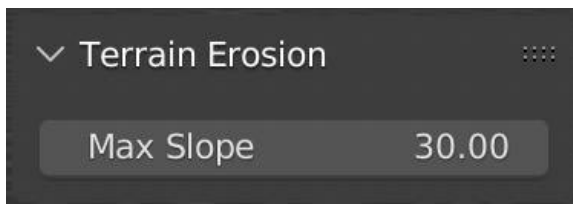
Prístup k inicializovanému objektu `erosion_properties` z metódy `draw(self, context: bpy.types.Context)` definovanej v triede panelu je možný cez kontext, ktorý je predaný ako parameter (výpis 5.5).

```
properties: ErosionProperties = context.scene.erosion_properties
row = self.layout.row()
row.prop(properties, "th_max_slope")
```

Výpis 5.5: Vytvorenie UI prvku, ktorý reprezentuje objekt typu `property`.

Výpis 5.5 taktiež obsahuje kód pre zobrazenie objektu `property` v užívateľskom rozhraní. Slúži na to metóda UI kontajneru, s názvom `prop` s parametrami `data` (dáta z ktorých sa má získať `property`) a `property` (identifikátor `property` v dátach).

Vytvorený objekt je zobrazený v paneli na obrázku 5.8.



Obrázek 5.8: Panel s UI prvkom pre ovládanie property s názvom „Max Slope“.

5.3.2 Operátory

Operátory sú objekty, ktoré vykonávajú operácie. Každá operácia, ako napríklad vytvorenie objektu, je v Blenderi vykonávaná operátorom. Erózie implementované v kapitolách 5.5, 5.6 a 5.7 sú tiež vykonávané operátorom.

Nový operátor je možné vytvoriť podobným spôsobom ako objekty typu property 5.3.1, definovaním triedy ktorá dedí z triedy `bpy.types.Operator` (výpis 5.6). Potrebné je v tejto triede definovať atribúty `bl_label` a `bl_idname`, ktoré reprezentujú názov a ID operátora.

```
class TestOperator(bpy.types.Operator):
    bl_label = "Test"
    bl_idname = "wm.test_operator"

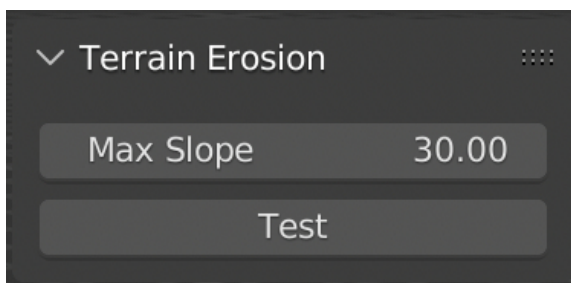
    def execute(self, context):
        print("This is a test message.")
        return {'FINISHED'}

...

# draw() method of Panel
row = self.layout.row()
row.operator("wm.test_operator")
```

Výpis 5.6: Definícia nového operátora, ktorý po spustení vypíše do konzole testovaciu správu.

Pomocou ID operátora je možné operátor vložiť ako tlačidlo do užívateľského rozhrania (obrázok 5.9), ktoré stlačením zavolá funkciu `execute()` definovanú v operátore (výpis 5.6).



Obrázok 5.9: Panel s tlačidlom pre spustenie operátora „Test“.

5.4 Vytváranie terénu

Súčasťou zásuvného modulu je operátor pre vytváranie terénu. Terén generovaný šumovými funkciami (kapitola 5.4.2) je reprezentovaný výškovou mapou a pre použitie v Blenderi, konvertovaný na MESH objekt.

5.4.1 Reprezentácia MESH objektov

Mesh objekty sú jedným z typov objektov dostupných v Blenderi. Jedná sa o sieť bodov poskladaných do n-uholníkov (polygonov).

BMVert

Typ `BMVert` reprezentuje jeden bod v 3D priestore. Obsahuje atribút `co` typu `mathutils.Vector` reprezentujúci súradnice daného bodu a sprístupňuje jednotlivé zložky cez atribúty s názvami osí `x`, `y`, `z` a iných, ktoré sú v tomto prípade nevyužité. Definuje bežné operácie ako sčítanie, alebo násobenie medzi objektami tohoto typu. Taktiež definuje atribút `link_edges` obsahujúci zoznam hrán `BMEdge` ktorých je daný bod súčasťou.

BMEdge

Typ `BMEdge` reprezentuje hranu medzi dvomi bodmi typu `BMVert`. Modul `BMesh` zabezpečuje, že neexistuje viacero bodov `BMVert` s rovnakými súradnicami, to znamená že susedné hrany vždy zdieľajú spoločné body, ktoré sú totožný objekt.

Poskytuje aj funkciu `other_vert(vert: BMVert) -> BMVert`, ktorej výstupom je druhý bod hrany, kde parameter `vert` je nám známy bod.

BMFace

Typ `BMFace` reprezentuje stenu, polygon, a skladá sa z 3, alebo viac bodov `BMVert` a hrán `BMEdge`.

BMesh

Typ `BMesh` reprezentuje celý objekt typu `MESH`, pozostávajúci z kolekcií objektov typu `BMVert`, `BMEdge`, `BMFace`, dostupných v atribútoch `verts`, `edges` a `faces`

```
for v in range(my_bmesh.verts):
    for le in v.link_edges:
        # Neighbouring vertex
        v_neigh = le.other_vert(v)
```

Výpis 5.7: Ukážka použitia objektu `my_bmesh` typu `BMesh`. Vonkajší cyklus iteruje cez všetky body objektu a vnútorný iteruje cez všetky hrany, ktorých je daný bod súčasťou. Na poslednom riadku je získaný susedný bod `v_neigh` bodu `v` z hrany `le`.

5.4.2 Šumové funkcie z modulu `mathutils`

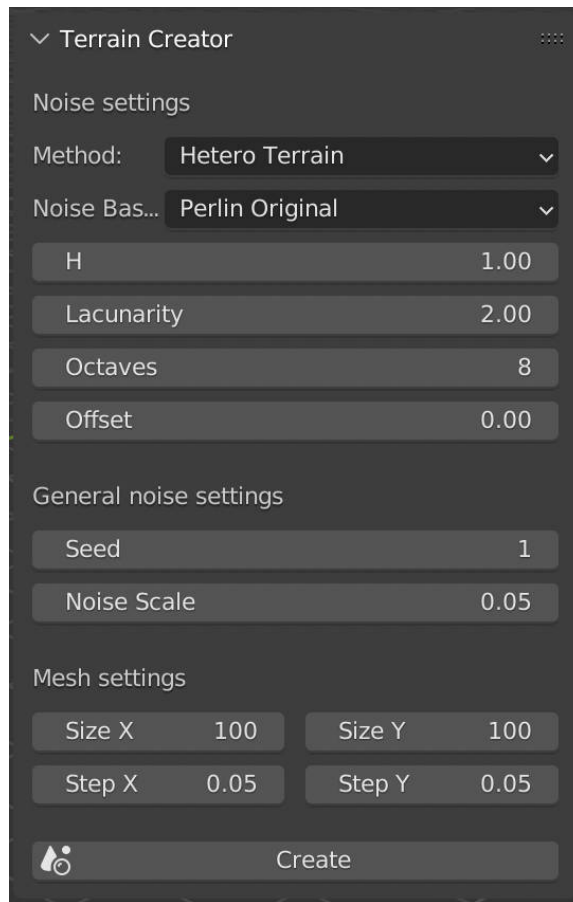
Python modul `mathutils` z Blender API poskytuje radu matematických funkcií. Patria medzi ne aj funkcie na generovanie šumu, ktorý su využité na procedurálne generovanie terénu v tomto operátore. Podmodul `mathutils.noise` obsahuje funkcie na generovanie rôznych druhov šumu a ich zoznam s možnými parametrami pre použitie je dostupný v dokumentácii ⁹.

5.4.3 Implementácia operátora pre generovanie terénu

Tento operátor využíva objekty typu `property` definované v triede `TerrainProperties` pre nastavovanie parametrov generovaného terénu. Patria medzi ne špecifické nastavenia šumu z modulu `mathutils` (kapitola 5.4.2), všeobecné nastavenia šumu a nastavenia objektu terénu.

⁹Šumové funkcie dostupné v module `mathutils`: <https://docs.blender.org/api/current/mathutils.noise.html>

Tieto nastavenia v podobe property objektov sú vložené do Panelu pre vytváranie terénu (obrázok 5.10).



Obrázok 5.10: Panel pre vytváranie terénu. Obsahuje špecifické nastavenia vybraného šumu, všeobecné nastavenia šumu ako `seed` a škála šumu a nastavenia objektu terénu, jeho rozmery (počet bodov) a hustota bodov.

Stlačením tlačidla „Create“ sa zavolá tento operátor, ktorý vytvorí terén s danými parametrami. Vo výpise 5.8 je ukázaný proces vytvárania objektu. Najskôr je vytvorený nový objekt a jeho mesh a je pridaný do Blender scény. Potom nasleduje proces generovania výškovej mapy s využitím šumových funkcií, ukladajú sa body, reprezentované trojicou desatinných čísiel, do pola `vertices` a ich prepojenia v podobe polygónov, n-tíc indexov bodov, v poli `polygons`. V poslednom kroku sú tieto polia použité pre naplnenie mesh objektu pomocou metódy `from_pydata()`.

```
# Create object and its mesh
terrain_mesh: bpy.types.Mesh = bpy.data.meshes.new("Terrain")
terrain_obj: bpy.types.Object = bpy.data.objects.new("Terrain", terrain_mesh)

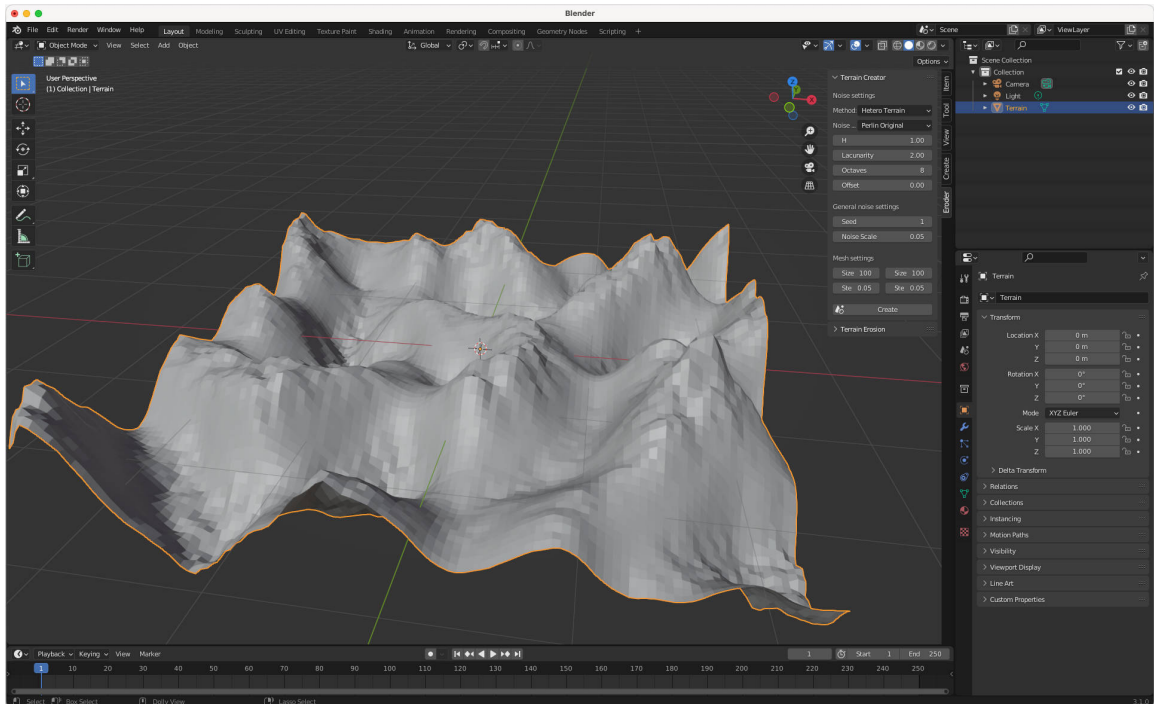
# Add object to scene
view_layer = bpy.context.view_layer
view_layer.active_layer_collection.collection.objects.link(terrain_obj)
```

```

...
# Fill mesh with polygons
terrain_mesh.from_pydata(vertices, [], polygons)

```

Výpis 5.8: Vytvorenie nového objektu, prídanie do scény a naplnenie objektu typu mesh polygónmi.



Obrázek 5.11: Terén vytvorený pomocou nastavení zobrazených na obrázku.

5.5 Termálna erózia

Termálna erózia je implementovaná v súbore `thermal_erosion.py`. Jej vstupom sú objekt typu `BMesh` a nastavenia typu `ErosionSettings` (výpis 5.9).

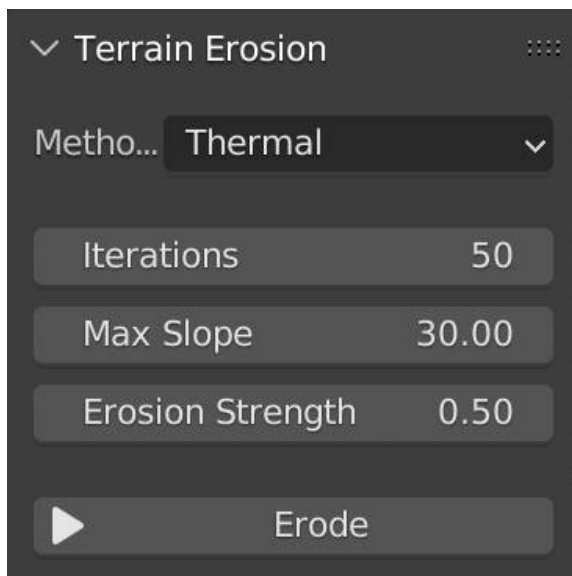
```

class ThermalErosionSettings:
    iterations: int
    max_slope: float
    erosion_strength: float

```

Výpis 5.9: Nastavenia termálnej erózie. Atribút `iterations` je počet iterácií, `max_slope` je maximálny uhol (talus) a `erosion_strength` udáva rýchlosť erózie v rozsahu 0 až 1, kde 0 znamená že sa žiaden svah nezosúva a 1 znamená že sa zosúva celý rozdiel výšky. Vysoké hodnoty spôsobujú nestabilitu, odporúčaná hodnota je 0.5.

Implementovaná metóda na rozdiel od metódy popísanej v kapitole 3.2 pracuje aj s uhlami svahu, pretože nejde zaručiť že erodovaný objekt má rovnomerne vzdialené body a preto nemôže byť ich rozdiel výšok používaný ako hlavný faktor rozhodujúci o erózii.



Obrázek 5.12: Panel erózie s vybratou metódou termálnej erózie a jej parametre.

5.6 Vodná erózia pomocou celulárneho automatu

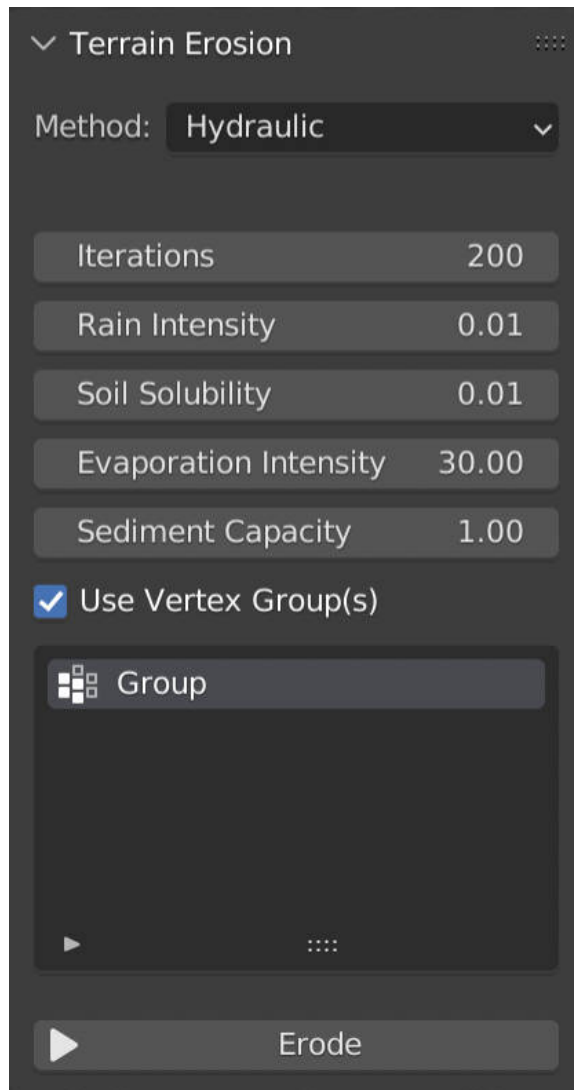
Vodná erózia implementovaná pomocou celulárneho automatu je založená na metóde popísanej v kapitole 3.3.1 a je umiestnená v súbore `thermal_erosion.py`. Nastavenia tejto metódy sú početnejšie než v termálnej erózii (kapitola 5.5), sú definované v triede `HydraulicErosionSettings` (výpis 5.10).

```
class HydraulicErosionSettings:
    iterations: int
    rain_intensity: float
    soil_solubility: float
    evaporation_intensity: float
    sediment_capacity: float
    # Indices of selected vertices of vertex groups
    # (if this option is selected)
    selected_vertex_indices: list[int] = None
```

Výpis 5.10: Nastavenia vodnej erózie (implementácia celulárnum automatom). Atribút `iterations` je počet iterácii, `rain_intensity` je množstvo zrážok, ktoré pribudne každou iteráciou na každý bod terénu, `soil_solubility` reprezentuje rozpustnosť terénu, `evaporation_intensity` udáva koľko percent objemu vody sa každú iteráciu vyparí a `sediment_capacity` udáva kapacitu sedimentu v jednotke vody. Navyše, `selected_vertex_indices` môže obsahovať zoznam indexov bodov, ktorý vymedzuje zrážky na body s vybratým indexom.

5.7 Vodná erózia pomocou simulácie častíc

Implementácia tejto metódy stavia na teórii popísanej v kapitole 3.3.2, ale niektoré časti boli zjednodušené pre jednoduchšiu implementáciu. Implementovaný algoritmus, umiestnený v súbore `hydraulic_erosion_pb.py`, posúva simulované kvapky iba po bodoch terénu, nie v

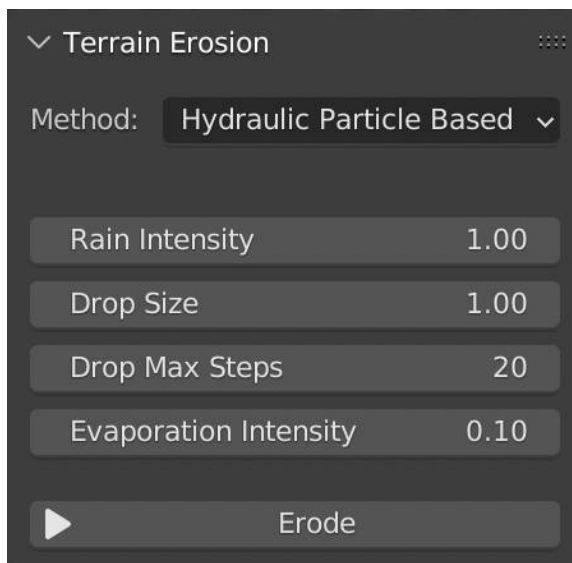


Obrázek 5.13: Panel erózie s vybratou metódou vodnej erózie (celulárny automat) a jej parametre.

priestore medzi nimi. Zároveň nepracuje s parametrom p_{radius} , ktorý v popísanej metóde v teórii slúži na ovládanie polomeru vplyvu kvapky. Tieto zjednodušenia majú za následok nepresvedčivý výzor erodovaného terénu, na ktorý je poukázané v kapitole testovania 6. Výpis 5.11 ukazuje nastavenia tejto metódy definované v triede `HydraulicErosionPBSettings`.

```
class HydraulicErosionPBSettings:
    rain_intensity: float
    drop_size: float
    drop_max_steps: int
    drop_evaporation_intensity: float
```

Výpis 5.11: Nastavenia vodnej erózie (implementácia simuláciou častíc) definuje atribút `rain_intensity` určujúci prínos zrážok (počet kvapiek pre každý bod terénu), `drop_size` je objem kvapky, `drop_max_steps` je maximálna dĺžka cesty jednej kvapky a `drop_evaporation_intensity` vyjadruje koľko percent kvapky sa vyparí po každom kroku.



Obrázek 5.14: Panel erózie s vybratou metódou vodnej erózie (simulácia častíc) a jej parametre.

5.8 Zobrazenie postupu erózie

Po spustení operátora erózie, program Blender prestane reagovať a „zamrzne“. Je to spôsobené tým že sa výpočet vykonáva v hlavnom vlákne, ktoré je zodpovedné aj za vykresľovanie užívateľského rozhrania.

Pre vyriešenie tohoto problému bolo potrebné presunúť výpočet do vedľajšieho vlákna. V operátore `ErosionOperator` je pri aktivácii vytvorené nové vlákno do ktorého sa presunie všetok výpočet simulácie. Hlavné vlákno je následne prenechané Blenderu pre pokračovanie vo vykresľovaní UI. Je to dosiahnuté metódou `invoke()`, ktorá vracia hodnotu `{"RUNNING_MODAL"}`, čím dá Blenderu vedieť, že panel beží v modálnom móde, teda že akcia beží na pozadí. Blender zaistí že sa v blízkej dobe zavolá metóda `modal()` ktorá signalizuje či je operácia hotová, alebo nie (výpis 5.12).

```
def modal(self, context: bpy.types.Context, event: bpy.types.Event):
    if event.type in ["RIGHTMOUSE", "ESC"]:
        self.stop_erosion(context, abort=True)
        return {'CANCELLED'}
    elif event.type in ['TIMER']:
        self.update_progress(context)
        self.check_thread()

        if not self.erosion_status.is_running:
            self.stop_erosion(context)
            return {"FINISHED"}

    return {"PASS_THROUGH"}
```

Výpis 5.12: Funkcia `modal()` definovaná v operátore erózie. Pokiaľ nie je výpočet hotový, vráti hodnotu `{"PASS_THROUGH"}` a jej volanie sa bude opakovať. V prípade že sa výpočet

skončil, vráti hodnotu {"FINISHED"}, čím sa definitívne skončí vykonávanie akcie operátora. Poslednou možnosťou je, že užívateľ stlačí klávesu ESC, alebo pravé tlačidlo myši a táto funkcia vráti hodnotu {"CANCELLED"}, čím signalizuje že sa má akcia operátora zrušiť.

Stav vlákna v ktorom je spustená simulácia je signalizovaný pomocou objektu triedy `ErosionStatus`, zobrazený na výpise 5.13.

```
class ErosionStatus:
    is_running: bool
    # signalize that the erosion (usually in other thread) should stop
    stop_requested: bool = False
    progress: int
```

Výpis 5.13: Trieda reprezentujúca stav vlákna, ktoré vykonáva simuláciu erózie. Atribút `is_running` signalizuje či je simulácia ešte vykonávaná, `stop_requested` slúži na poslanie správy z hlavného vlákna vedľajšiemu, aby ukončilo výpočet, keď to bude možné a `progress` je percento dokončenej práce v simulácii. Hodnota `progress` sa posiela z vlákna simulácie do hlavného vlákna, ktoré môže aktualizovať hodnotu property, ktorá aktualizuje užívateľské rozhranie.

Informácie o stave erózie sú komunikované pomocou objektov property, napríklad definícia property reprezentujúcu postup erózie je zobrazená na výpise 5.14.

```
# class ErosionProperties

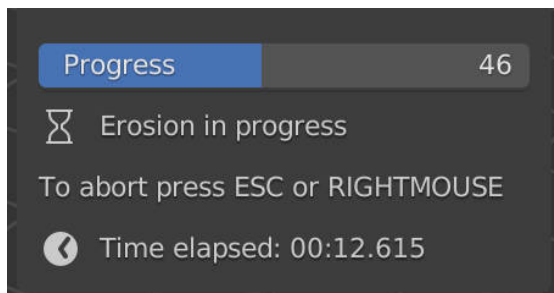
def __update_ui(self, context: bpy.types.Context):
    region: bpy.types.Region
    for region in context.area.regions:
        if region.type == "UI":
            region.tag_redraw()
    return None

progress: IntProperty(
    name="Progress",
    default=0,
    min=0,
    max=100,
    description="Progress of the running erosion",
    update=__update_ui
)
```

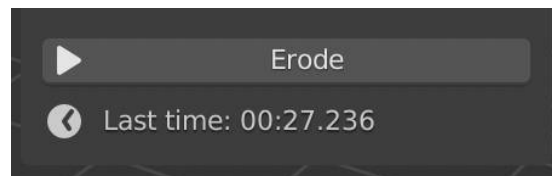
Výpis 5.14: Stav erózie reprezentovaný v property `Progress`. Po zmene hodnoty je zavolaná metóda `__update_ui()`, ktorá vynúti prekreslenie užívateľského rozhrania, čím zaručí že sa nová hodnota zobrazí v paneli erózie.

5.9 Porovnanie výsledkov implementovaných metód

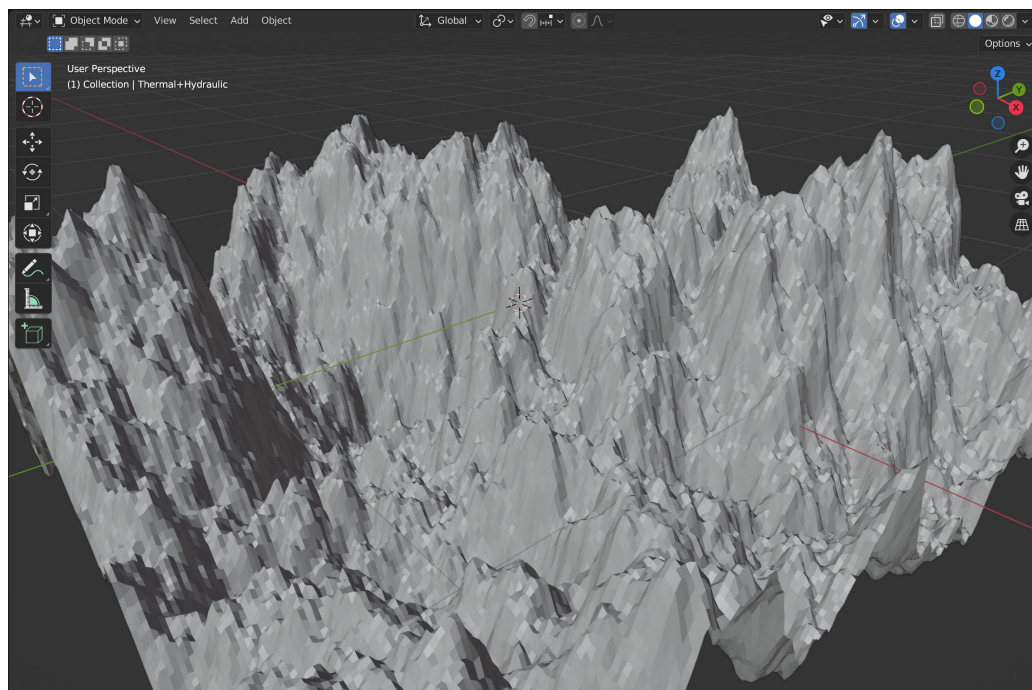
Táto kapitola obsahuje obrázky výsledkov, dosiahnutých implementovanými metódami erózie.



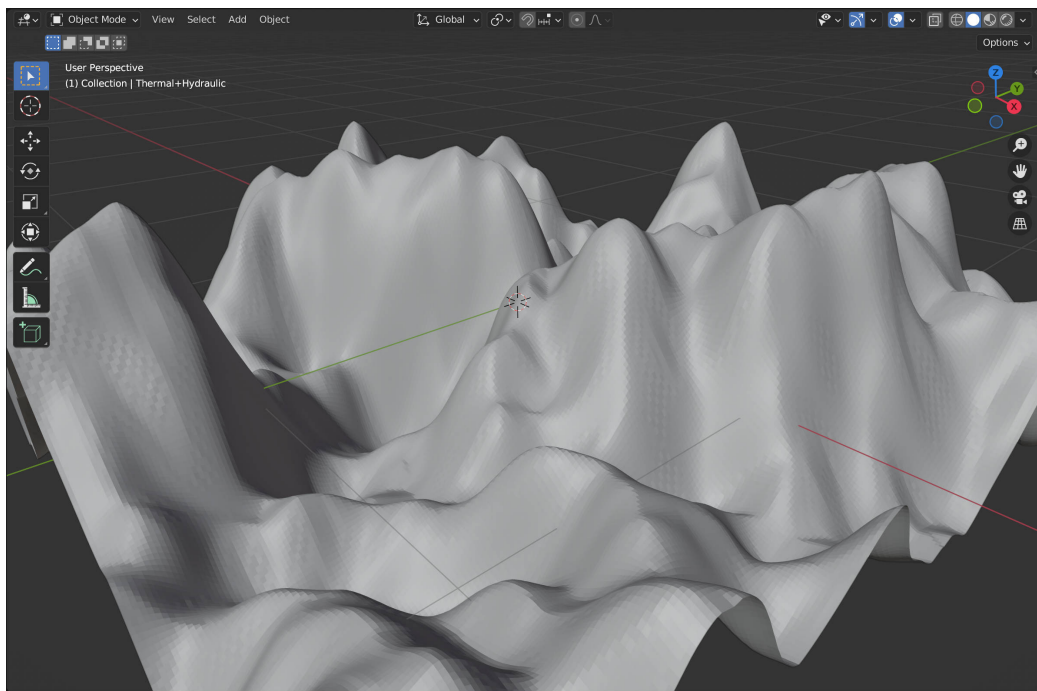
Obrázek 5.15: Zobrazenie stavu erózie v paneli. Stav (progress bar) sa posúva podľa percentuálneho dokončenia erózie. Proces erózie sa dá zrušiť stlačením klávesy ESC alebo pravým tlačidlom myši. Úplne dole je zobrazený ubehnutý čas erózie



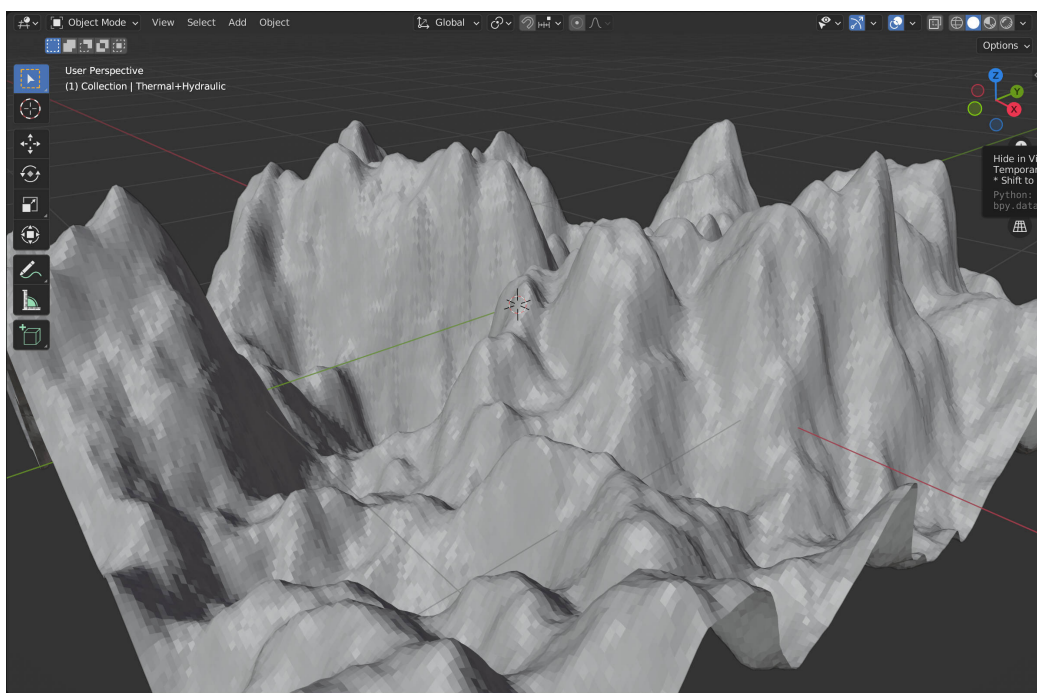
Obrázek 5.16: Po dokončení erózie je zobrazený celkový čas výpočtu pod tlačidlom „Erode“.



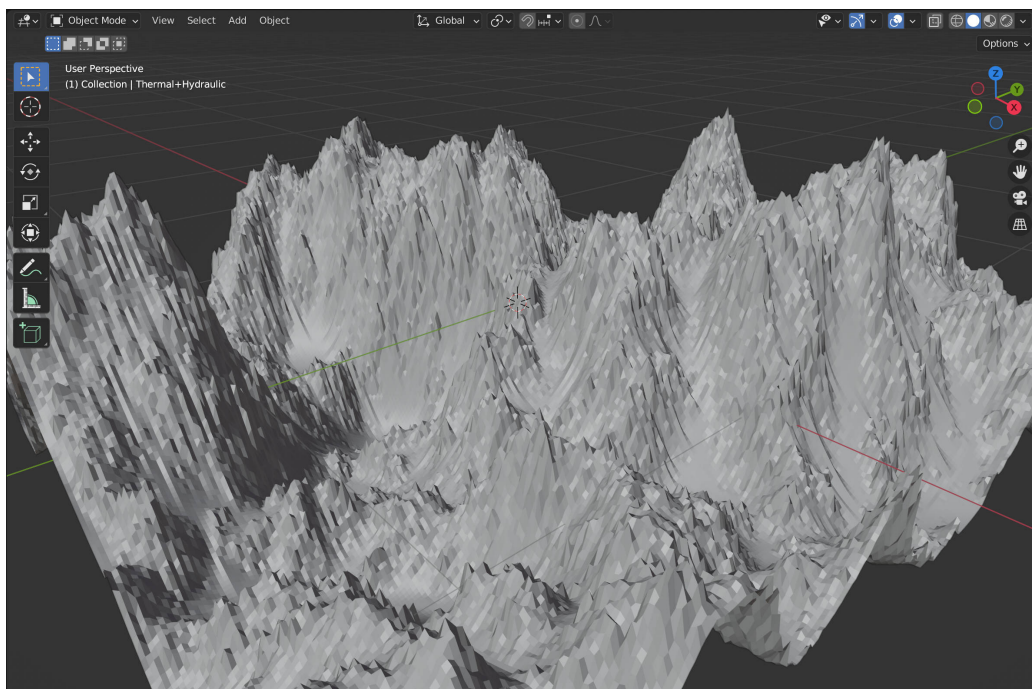
Obrázek 5.17: Originálny terén použitý pre porovnanie metód. Zmenené prednastavené hodnoty: Method = Fractal, Seed = 4, Noise Scale = 0.01, Size X = Size Y = 300 a Step X = Step Y = 0.01.



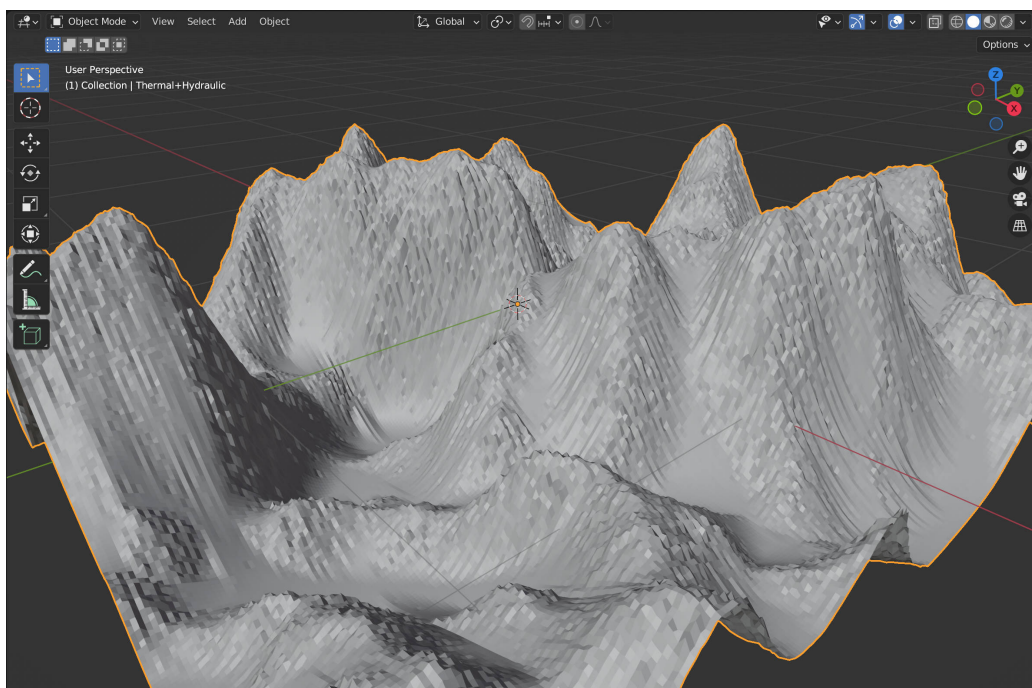
Obrázek 5.18: Výsledok po termálnej erózii. Zmenené prednastavené hodnoty: Erosion Strength = 30.



Obrázek 5.19: Výsledok po vodnej erózii (celulárny automat). Zmenené prednastavené hodnoty: Iterations = 20, Rain Intensity = 0.1, Soil Solubility = 0.5 a Evaporation Intensity = 10.



Obrázek 5.20: Výsledok po vodnej erózii (simulácia častíc). Zmenené prednastavené hodnoty: Rain Intensity = 2, Drop Size = 10, Drop Max Steps = 100 a Evaporation Intensity = 0.01.



Obrázek 5.21: Výsledok po termálnej a následne vodnej erózii (simulácia častíc). Boli použité parametre zmienené v popise obrázka 5.18 a 5.20.

Kapitola 6

Testovanie

Výsledný zásuvný modul bol testovaný na vzorke 5 respondentov. Cieľom bolo zistiť či je aplikácia intuitívna a jednocuhá na použitie, či jej funkcionálna vykonáva to čo by očakávali a či by tento zásuvný modul niekomu odporúčali.

Všetci 5 respondenti reagovali kladne ohľadom prvej témy. Pochvalovali si jednoduchý proces inštalácie a prívetivé používateľské rozhranie. Jeden respondent, ktorý nebol skúsený s programom Blender, mal potiaže nájsť modul v užívateľskom rozhraní, pretože panel bol pri štarte programu schovaný. Avšak po prečítaní návodu sa dozvedel, ako zobrazit tento panel.

Ich obľúbenou časťou UI bol indikátor postupu s časovačom, vďaka ktorému mali vždy prehľad nad stavom simulácie.

V druhej téme, ktorá sa venuje funkcionálite programu, neboli až tak pozitívny. Aj keď s metódou termálnej erózie boli všetci spokojný, vedeli ako ju ovládať aby dosiahli žiadaný efekt, s metódami vodnej erózie neboli veľmi spokojní.

Prvá metóda vodnej erózie, implementovaná pomocou celulárneho automatu je najväčším sklamaním, pretože jej simulácia trvá najdlhšie a jej efekt je len minimálny. V teréne sa netvorili takmer žiadne rysy odpovedajúce vodnej erózii, aj keď vyskúšali rôzne kombinácie parametrov.

Druhá metóda, implementovaná pomocou simulácie častíc bola o niečo lepšia. Respondentom sa pomocou tejto metódy podarilo vytvorit viditeľné žľaby a údolia v teréne a nánosy sedimentu, ale ich výzor nebol veľmi realistický. Problémom bolo, že žľaby boli príliš tenké na terénoch s vyššou hustotou bodov. Tento problém je spôsobený tým, že simulované kvapky berú do povedomia iba body v jej priamom okolí. Ďalším problémom bolo to, že žľaby neboli orientované kolmo na svah, čím vyzerali neprirodzene. Je to spôsobené tým, aká je štruktúra terénu (pravidelná mriežka bodov) a tým že kvapka nerozdeľuje jej objem ekvivalentne do okolitých bodov pri jej ceste dole kopcom. Modely s nižšou hustotou bodov vyzerali po erózii lepšie.

Respondenti boli spokojný s rýchlosťou tejto metódy a jednoduchosťou ovládania simulácie pomocou parametrov.

S panelom na vytváranie terénu boli veľmi spokojný. Najviac si pochvalovali preddefinované nastavenia rôznych šumových metód, vďaka čomu vedeli ihneď vytvorit zaujímavý terén.

K poslednej téme sa 2 vyjadrili neutrálne a 3 pozitívne. Konsenzus bol, že vytvorený zásuvný modul je použiteľný, ale má svoje limitácie. Tento modul je bezplatný takže si potenciálny nový používateľ môže jednoducho vyskúšať, či mu jeho funkcionálna vyhovuje.

Kapitola 7

Záver

Cieľom práce bolo vytvoriť zásuvný modul pre Blender, ktorý umožňuje erodovať modely terénu. Výsledný modul ponúka eróziu viacerými metódami a aj nástroj pre generovanie terénu a bol zverejnený pre verejnosť pre voľné použitie ¹.

Podarilo sa implementovať metódy termálnej a vodnej erózie. Termálnu eróziu je jednoduché nastaviť pre dosiahnutie požadovaného efektu vyhladenia terénu. V ponuke sú aj 2 implementácie vodnej erózie, implementácia využitím celulárneho automatu a implementácia pomocou simulácie častíc. Každá metóda má sadu špecifických parametrov simulácie, ktoré môže užívateľ konfigurovať a tým ovplyvňovať výsledný efekt erózie.

Prvá metóda vodnej erózie navyše umožňuje vymedziť plochu, na ktorú dopadajú zrážky, ale ako užívateľské testovanie (kapitola 6) odhalilo, je táto metóda sklamaním, pretože nevytvára v teréne rysy spájané s vodnou eróziou, ako sú zľaby, údolia a nánosy sedimentu.

Druhá metóda vodnej erózie je použiteľnejšia. Je oveľa rýchlejšia, jednoduchšie konfigurovateľná a jej efekty je jednoznačne vidieť. Avšak aj tie nie sú dokonalé, pri teréne s vyššou hustotou bodov pôsobí výsledok erózie málo prirodzene.

Vytvorený zásuvný modul aj napriek svojim nedostatkom zjednodušuje prácu vytvárania terénu priamo v programe Blender, doplnením chýbajúcej funkcionality.

Ďalšia práca na tomto module by mohla pridať ďalšie metódy erózie, či už veternú eróziu, alebo eróziu vhodnú pre objekty iné než terén. Taktiež by sa mohli vylepšiť metódy vodnej erózie, ktoré neprodukujú žiadaný efekt a metódy založené na celulárnom automate by mohli byť zrýchlené optimalizovaním a paralelizáciou.

¹Zdrojové kódy zásuvného modulu dostupné online: <https://github.com/petak5/BP>

Literatura

- [1] BENES, B. a FORSBACH, R. Visual Simulation of Hydraulic Erosion. *WSCG*. 2002, sv. 10, 1-2, s. 79–86. ISSN 1213-6964. Dostupné z: http://wscg.zcu.cz/wscg2002/Papers_2002/F23.pdf.
- [2] BEYER, H. T. *Implementation of a method for hydraulic erosion*. Munich, DE, 2015. Bakalárska práca. Technische Universität München, Department of Informatics. Dostupné z: <https://www.firespark.de/resources/downloads/implementation%20of%20a%20methode%20for%20hydraulic%20erosion.pdf>.
- [3] BLENDER. *Blender Python API Documentation* [online]. 2022 [cit. 2022-5-16]. Dostupné z: <https://docs.blender.org/api/current/index.html>.
- [4] BLENDER. *Blender.org* [online]. 2022 [cit. 2022-5-15]. Dostupné z: <https://www.blender.org/>.
- [5] MEI, X., DECAUDIN, P. a HU, B.-G. Fast Hydraulic Erosion Simulation and Visualization on GPU. In: *15th Pacific Conference on Computer Graphics and Applications (PG'07)*. 2007, s. 47–56. DOI: 10.1109/PG.2007.15. ISBN 1550-4085. Dostupné z: <https://ieeexplore.ieee.org/document/4392715>.
- [6] MUSGRAVE, F. K., KOLB, C. E. a MACE, R. S. The Synthesis and Rendering of Eroded Fractal Terrains. *SIGGRAPH Comput. Graph.* New York, NY, USA: Association for Computing Machinery. jul 1989, sv. 23, č. 3, s. 41–50. DOI: 10.1145/74334.74337. ISSN 0097-8930. Dostupné z: <https://doi.org/10.1145/74334.74337>.
- [7] OLSEN, J. Realtime Procedural Terrain Generation. In: . 2004. Dostupné z: <http://web.mit.edu/cesium/Public/terrain.pdf>.
- [8] TÓTH, B. Fast Hydraulic and Thermal Erosion on the GPU. In: . 2011. Dostupné z: <https://old.cescg.org/CESCG-2011/papers/TUBudapest-Jako-Balazs.pdf>.
- [9] ŠT'AVA, O., BENEŠ, B., BRISBIN, M. a KŘIVÁNEK, J. Interactive Terrain Modeling Using Hydraulic Erosion. In: *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Goslar, DEU: Eurographics Association, 2008, s. 201–210. SCA '08. ISBN 9783905674101. Dostupné z: <https://dl.acm.org/doi/pdf/10.5555/1632592.1632622>.