

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Procedurální generování textur

Bakalářská práce

Autor: Pavlína Kuřeová
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Karel Petránek

Hradec Králové

duben 2016

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 19.4.2016

Pavλίna Kuřeová

Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Karlu Petránkovi za metodické vedení a rady při psaní práce.

Anotace

Bakalářská práce zkoumá a shrnuje metody procedurálního generování textur používané v oblasti počítačové grafiky. Popisovanými metodami jsou: Perlinova šumová funkce, fraktály, buněčné texturování a L-systémy. Dále se zaměřuje na představení existujících aplikací založených na zkoumaných metodách, pomocí kterých mohou být procedurální textury generovány.

Práce popisuje principy metod procedurálního generování textur a jejich následné použití v představených aplikacích, které umí jednotlivé metody generovat, jejich popis, doporučení, zhodnocení a předvedení vygenerovaných textur.

Annotation

Title: Procedural Texture Generation

This Bachelor Thesis describes useful methods for generating procedural textures in computer graphics. The following methods are described: Perlin noise, Fractals, Cellular Texturing and L-systems. Furthermore, existing programs for procedural texture generation based on the above methods are presented.

This thesis describes the principles of the above methods, presents their use in existing programs which use these methods to generate textures. Existing programs are analyzed both on objective and subjective measures.

Klíčové výrazy: procedurální, generování, textura, Perlinův, šum, turbulence, fraktál, buněčné texturování, l-systém

Key words: procedural, generation, texture, Perlin, noise, turbulence, fractal, cellular texturing, l-system

Obsah

1	Úvod	1
2	Textura	2
2.1	Pojem procedurální textura	3
3	Metody pro generování procedurálních textur	4
3.1	Perlinova šumová funkce	4
3.1.1	Skládání šumových funkcí.....	6
3.1.1.1	Persistence.....	7
3.1.2	Turbulence	9
3.1.3	Simplexová šumová funkce	9
3.1.4	Vylepšení Perlinovy šumové funkce	11
3.2	Procedurální generování textur pomocí fraktálů	13
3.3	Buněčné texturování.....	14
3.4	Procedurální generování pomocí Lindenmayerových systémů	16
3.4.1	Želví grafika	19
3.4.2	Závorkové L-systémy	20
4	Testování metod a hodnocení vybraných aplikací.....	22
4.1	Perlinova šumová funkce	22
4.1.1	Perlin Noise Generator	22
4.1.2	TextureGenerator-Online	23
4.1.3	Filter Forge.....	24
4.2	Fraktály	27
4.2.1	Mystica.....	27
4.2.2	Ultra Fractal	30
4.3	Buněčné texturování.....	33
4.3.1	Worley Texture Generator	33

4.3.2	Filter Forge.....	35
4.4	L-systémy	35
4.4.1	Lindenmayer System Generator	36
4.4.2	L-Systems Explorer	37
4.4.3	WWW L-System Explorer	40
4.4.4	L-Studio	42
5	Shrnutí výsledků hodnocení	46
6	Závěr	48
7	Seznam použité literatury	49
8	Seznam obrázků	54
9	Seznam tabulek.....	57

1 Úvod

Procedurálně generované textury (procedurální textury) nejsou reprezentovány rastrovým obrázkem, ale matematickou funkcí (procedurou), která každému bodu ležícímu v dvojrozměrném či trojrozměrném prostoru přiřazuje jeho barvu.

Takové textury se nejčastěji používají ke generování přírodních a fraktálních textur typu dřevo, mramor, kamení či oblaka s téměř neomezeným rozlišením. Pomocí L-systémů se generují textury rostlin.

Při použití procedurálních textur nedochází k nežádoucímu *aliasu* způsobenému příliš nízkým rozlišením, ke kterému při mapování klasických textur běžně dochází. *Alias* je vada v navzorkovaném obrazu (artefakt) způsobená podvzorkováním a *aliasing* je pojem pro vznik artefaktů v důsledku vzorkování¹. Podvzorkování způsobuje zobrazení vysokých frekvencí do nízkých a tak vznikají vizuálně rušivé artefakty (angl. *jaggies a efekt moire*). Pro odstranění těchto nežádoucích artefaktů se obraz musí vzorkovat s minimálně 2x vyšší frekvencí, než je nejvyšší frekvence v datech [Proakis et al., 1996]. Takovému potlačování nesrovnalostí způsobených podvzorkováním se obecně říká *antialiasing*. Tento pojem je rozebrán v [Tanner et al., 1989].

Další výhodou procedurálních textur jsou nízké nároky na kapacitu operační paměti, protože procedurální textura se nevytváří dopředu (není uložena ani v operační paměti, ani v externím souboru), barva jednotlivých bodů na povrchu těles se počítá až při renderování².

Vhodně definovaná procedura s parametry dokáže popsat široký rozsah textur prostřednictvím variace těchto parametrů. Realistická scéna může obsahovat stovky i tisíce různých textur, a proto je pro texturování objektů vhodné použít algoritmus namísto ukládání velkého množství textur v operační paměti.

¹ Vzorkování je diskretizace signálu. Převod analogového signálu na digitální.

² Renderování je vykreslení dvourozměrného obrazu na základě geometrického popisu scény.

2 Textura

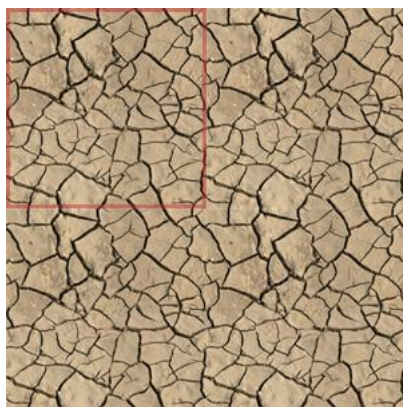
Textura popisuje vlastnosti povrchu a je důležitá pro vnímání struktury, barvy a typu objektu. Textury dodávají objektům, či prostředí vizuální realističnost a pomáhají určit jejich podstatu. Jeden model, např. koule, se může díky texturám změnit z obyčejného modelu hracího míče například na model slunce, aniž by se jakkoli manipulovalo s geometrií objektu. Klasické textury jsou předem připravené rastrové obrazy, které se posléze mapují na objekty. Je tedy nutné zvolit si správně velké rozlišení, aby textura nepůsobila roztaženě (moc malá textura mapovaná na moc velkou plochu).

Textury, které se velmi často používají, neboť nemají žádné znatelné přechody při dláždění ploch, se nazývají „bezešvé textury“. Používají se na větší plochy jako terén, hladina vody, zdi, atd. Bezešvými texturami a vylepšenou metodou na vytváření bezešvých textur se zabývá např. [Allene et al., 2008].



Obr. č. 1 Příklad bezešvé textury, rozpraskaná ztvrdlá hlína.

Zdroj: [Galleryhip]



Obr. č. 2 Čtyři stejné bezešvé textury vedle sebe vytvářejí větší plochu bez viditelného napojení textur.

2.1 Pojem procedurální textura

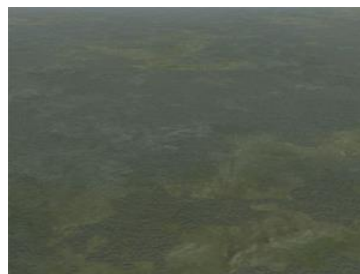
Bezešvé textury nejsou vždy vhodným řešením, jak docílit dobře vypadajícího texturování rozlehlých ploch (např. terénu) díky častému opakování stejného vzoru. Vhodným řešením tohoto problému jsou procedurální textury, které nevyužívají rastrové obrazy k pokrytí plochy objektu či terénu, ale matematickou funkci (proceduru), díky níž se dá snadno vygenerovat textura s následujícími vlastnostmi:

- minimální nároky na paměť
- funkce vzniklá na základě této techniky je spojitá, je možné vytvořit či vzorkovat texturu v libovolném rozlišení
- není periodická, můžeme pomocí ní pokrýt velké plochy bez obav z opakování vzorku

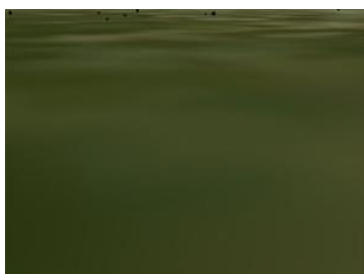
Třetí vlastnost řeší problém klasických textur – příliš časté opakování vzoru, což je na rozlehlých plochách dobře patrné, především u přechodů mezi různými rozlišeními textury. Nevýhodou procedurálních textur je, že ne všechny povrchy lze dostatečně jednoduše matematicky vyjádřit.



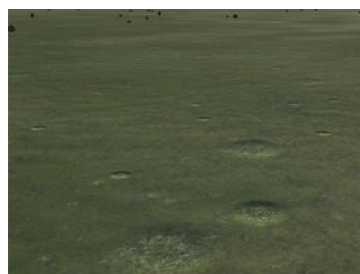
Obr. č. 3 Klasická textura, problém s přechody a opakováním.



Obr. č. 4 Použití procedurální textury, odstraňuje problém s přechody a opakováním.



Obr. č. 5 Klasická textura, problém s rozlišením.



Obr. č. 6 Použití procedurální textury, odstraňuje problém s rozlišením.

Zdroj: [Flightgear Wiki, 2014]

3 Metody pro generování procedurálních textur

Tato kapitola se zabývá popisem metod pro generování procedurálních textur. Tyto metody dokáží věrně napodobovat reálné povrchy a přenést je do textur, které se dále využívají, např. pro mapování na určitý model, k texturování terénu a mnoho dalších. Popisovanými metodami v této práci jsou:

- Perlinova šumová funkce
- generování pomocí fraktálů
- buněčné texturování
- generování pomocí L-systémů

Jednotlivé metody mají různá využití, některá se překrývají. Některé metody se na určité přírodní jevy hodí více než jiné, záleží vždy na cíli. Pokud je cílem vygenerovat například texturu dřeva či mramoru, je vhodnější použít Perlinovu šumovou funkci namísto L-systémů. K čemu jednotlivé metody slouží (či se k tomu nejvíce hodí), je popsáno níže.

3.1 Perlinova šumová funkce

Perlinova šumová funkce³ či Perlinův šum, angl. *Perlin noise*, je vhodný pro generování grafického šumu [Perlin, 1985]. Perlinova šumová funkce vnáší do světa textur náhodné prvky, které přibližují textury a následně i jimi texturované objekty blíže realitě.

Perlinova funkce je rychle vypočitatelná a má následující vlastnosti:

- funkce je statisticky invariantní vzhledem k otáčení
- je statisticky invariantní vzhledem k posunutí
- je spojitá
- má omezené frekvenční spektrum
- je opakovatelná, zavoláním funkce s parametrem x vrátí vždy stejnou hodnotu y

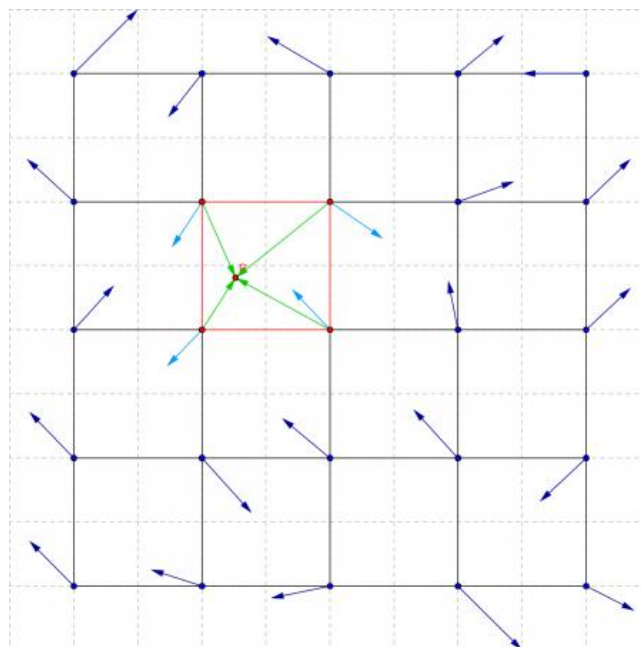
³ Tuto funkci představil Ken Perlin roku 1985. Vyvinul ji během prací pro Mathematical Applications Group, Inc. Ken Perlin původní šumovou funkci v roce 2002 vylepšil, potlačením některých vizuálních artefaktů [Perlin, 2002].

První dvě vlastnosti zajišťují, že se funkce statisticky ustálí, tedy nebude se měnit při posouvání a otáčení. Znamená to, že nebude záležet na místě, od kterého je textura mapována, ani na jejím natočení - textura bude statisticky soběpodobná.

Omezené frekvenční spektrum zaručuje, že funkce bude mít určitou maximální frekvenci, ale díky tomu by teoreticky mohla produkovat nežádoucí alias. Aliasu lze ale snadno zabránit zvolením nejvyšší frekvence podle přesnosti vzorkování. Ze seznamu vlastností je to především tato vlastnost, která urychluje výpočet hodnoty funkce.

Pátá vlastnost zaručuje, že volání funkce s daným parametrem vrátí jako výsledek vždy stejné číslo. Zavolá-li se funkce s určitou texturovací souřadnicí, bude mapovat na odpovídající místo vždy stejnou barvu [Žára et al., 2010].

Perlinův šum se řadí do kategorie přírůstkového šumu (angl. gradient noise), což znamená, že definuje pseudonáhodný vektor přírůstku (gradient) v pravidelně rozmístěných bodech (reprezentované celočíselnými souřadnicemi) v n-rozměrné mřížce [Gustavson, 2005]. Perlin navrhl použít pole vektorů o délce 256. K těmto vektorům se nepřístupuje přímo, ale pomocí pole náhodných permutací indexů, také o délce 256 [Žára et al., 2010].



Obr. č. 7 Pravidelná mřížka s náhodně směřovanými 2D vektory pro každý celočíselný bod v mřížce.

Zdroj: [ECG Wiki, 2011]

Šum v bodě (x, y) je určen vypočtením pseudonáhodného vektoru přírůstku (gradientu) na každém ze čtyř nejbližších vrcholů mřížky a následně je provedena interpolace [Perlin, 2002].



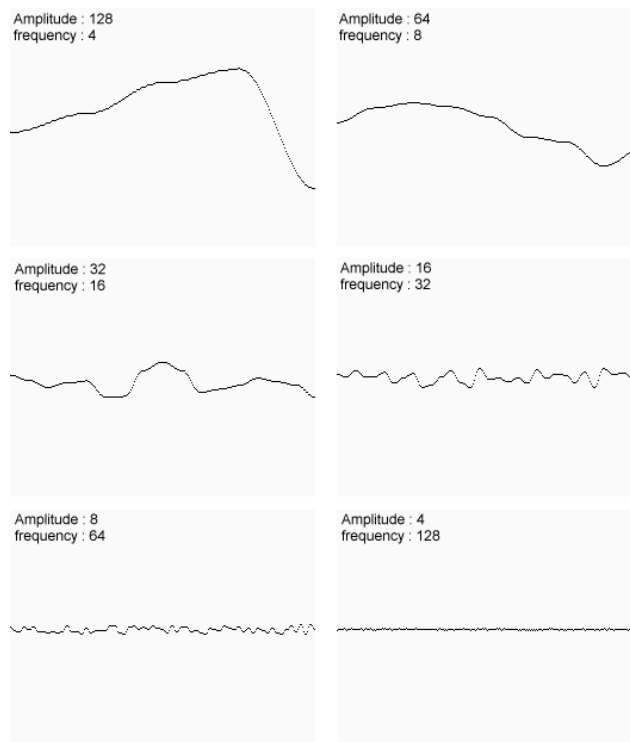
Obr. č. 8 Hladká interpolace mezi pravidelně uspořádanými body na ose (1D).

Zdroj: [Gustavson, 2005]

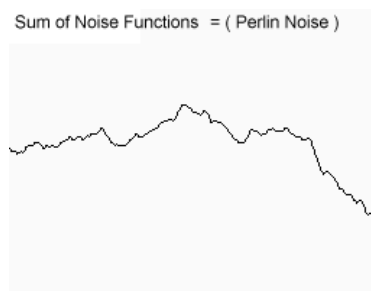
3.1.1 Skládání šumových funkcí

Složení šumových funkcí představuje součet funkcí, kdy má každá funkce změněnou amplitudu a frekvenci. Tyto dva matematické pojmy lze přiblížit na sinusoidě, kde amplituda představuje výšku vlny sinusoidy a frekvence představuje inverzní hodnotu jedné periody sinusoidy, tedy $1/T$, kde T značí periodu. Podrobnější rozbor těchto pojmů lze nalézt v [Reichl et al., 2006].

Výsledkem součtu šumových funkcí je šum, kterým je možné napodobit rozličné přírodní jevy, nejčastěji oblaka, vodu, dřevěné materiály a mramor. Obecně lze takto sčítat libovolné šumové funkce.

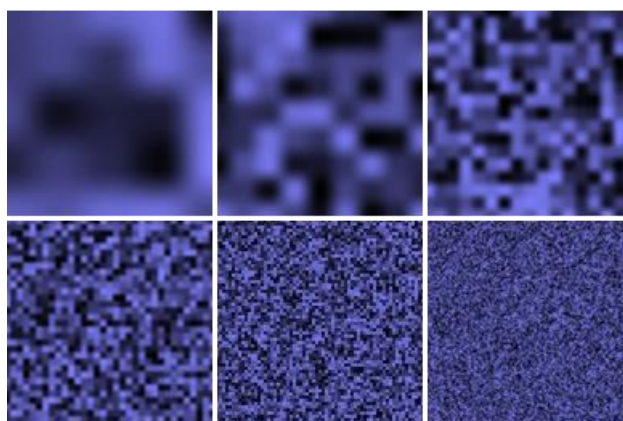


Obr. č. 9 Šumové funkce před součtem s různými hodnotami amplitud a frekvencí.

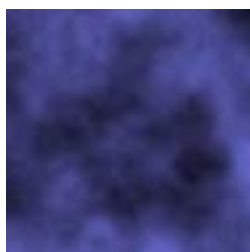


Obr. č. 10 Šumové funkce po součtu⁴.

Zdroj: [Hugo, 2000]



Obr. č. 11 Dvozměrné různé šumové funkce před součtem.



Obr. č. 12 Dvozměrná šumová funkce po součtu.

Zdroj: [Hugo, 2000]

3.1.1.1 Persistence

Persistence je pojem zavedený pro vyjádření amplitudy pro danou frekvenci. Čím vyšší je persistence, tím více se projevuje frekvence, tím zajímavější budou textury. Persistence určuje, jak rychle amplituda klesá mezi každou oktávou⁵. Zvý-

⁴ Často se takto získané funkce v literatuře přezdívá Perlinova funkce, i když se ve skutečnosti jedná o součet několika Perlinových funkcí se změněnou amplitudou a frekvencí [Žára et al., 2010].

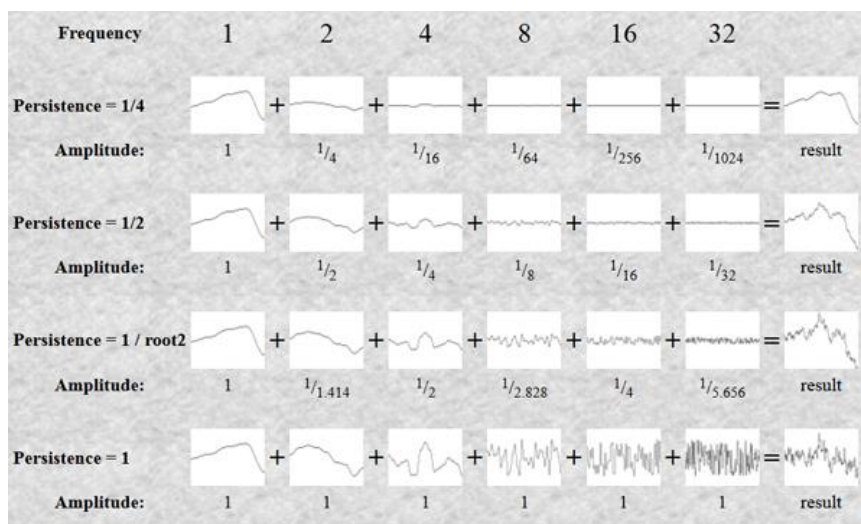
⁵ Oktáva představuje každou postupně přidanou šumovou funkci. Každá nově přidaná šumová funkce má dvojnásobnou frekvenci funkce předchozí.

šená míra persistence vytváří hrubší efekt na textuře, kdežto snížená míra persistence výslednou texturu zjemní.

$$\text{frekvence} = 2^i$$

$$\text{amplituda} = \text{persistence}^i$$

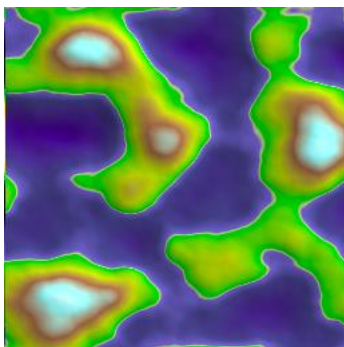
Kde i je i -tá přidaná šumová funkce [Hugo, 2000]. Pro ilustraci vlivu persistence na výstup Perlinovy šumové funkce slouží Obr. č. 13.



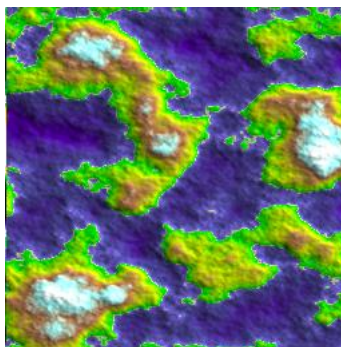
Obr. č. 13 Vliv persistence na výstup Perlinovy šumové funkce.

Zdroj: [Hugo, 2000]

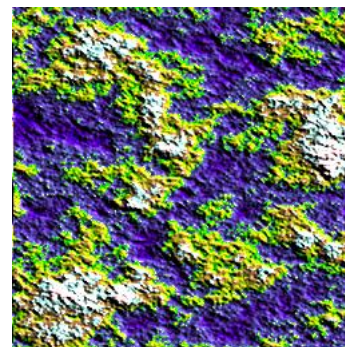
Pro ilustraci vlivu persistence na texturu slouží Obr. č. 14, Obr. č. 15 a Obr. č. 16. Na Obr. č. 14 je vidět, že povrch textury je mnohem hladší, obsahuje méně detailů, než Obr. č. 16.



Obr. č. 14 $p = 0.25$



Obr. č. 15 $p = 0.5$

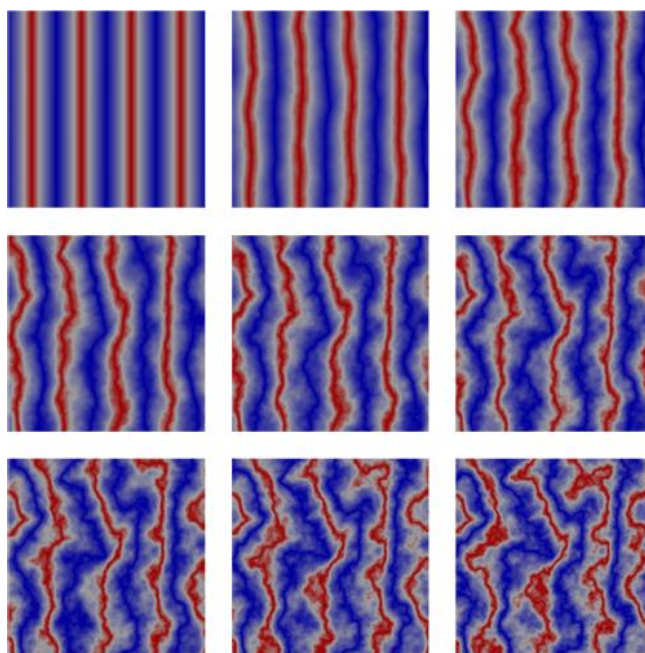


Obr. č. 16 $p = 0.75$

Zdroj: [Butram, 2015]

3.1.2 Turbulence

Pro vytváření různě zajímavých textur se na Perlinovu šumovou funkci aplikuje parametr zvaný *turbulence* [Tišnovský, 2008]. Díky změně tohoto parametru se výsledná funkce stává více náhodnou. Vyšší náhodnost je žádoucí parametr při generování procedurálních textur, protože zajišťuje větší realističnost textur. Pomocí turbulence lze vytvořit textury, které připomínají mramor, dřevo a další. Velikost turbulence ve funkci lze měnit podle potřeby, je to tedy parametr, jenž si uživatel zadává.



Obr. č. 17 Vlivy turbulence na původní pravidelné pruhy, vzniká textura mramoru.

Zdroj: [Tišnovský, 2008]

Jednotlivé povrchy na Obr. č. 17 se od sebe odlišují mírou vlivu turbulence. Zatímco u prvního povrchu je hodnota turbulence rovna 0, u povrchu posledního je to již 0,8.

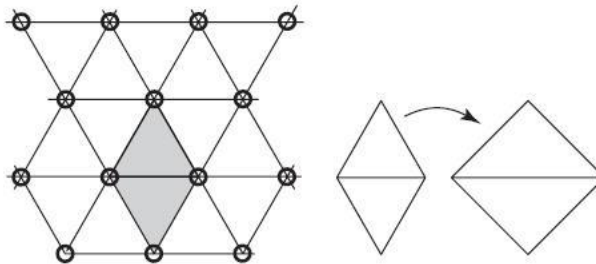
3.1.3 Simplexová šumová funkce

V roce 2001 Ken Perlin přišel se zjednodušením pro svou původní šumovou funkci. Simplexová šumová funkce (angl. Simplex noise) je srovnatelná s původní verzí Perlinovy funkce, pouze má za cíl odstranit některá omezení původní šumové funkce a to především pro N-dimenzionální šum [Gustavson, 2005].

Simplexová šumová funkce přináší tyto výhody:

- Je méně náročná na výpočet, vyžaduje méně násobení,
- nemá žádné patrné artefakty,
- má dobře definovanou a spojitou derivaci (gradient), která je zároveň jednodušší na výpočet,
- snadno implementovatelná pro hardware,
- výpočetně nenáročná i při vyšších dimenzích (4D, 5D a více).

Simplexový šum používá tzv. *simplex grid*, tedy jednoduchou souřadnicovou mřížku. Hlavní myšlenkou je najít co nejjednodušší a nejkompaktnější tvar, který lze opakovat a naplnit jím prostor s N dimenzemi. Pro jednorozměrný prostor jsou to intervaly stejné délky, které jsou kladeny jeden za druhým. V dvojrozměrném prostoru se používá rovnostranných trojúhelníků, které jsou skládány vedle sebe, a na sebe, což je znázorněno na Obr. č. 18.



Obr. č. 18 Ukázka tvaru pro 2D prostor.

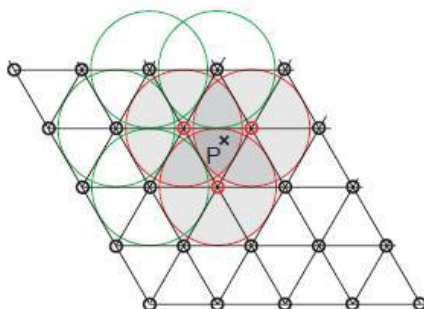
Zdroj: [Gustavson, 2005]

Obecně jde vždy o útvar, který má $N + 1$ vrcholů, kde N je počet dimenzí, a $N!$ útvarů může vyplnit N -dimenzionální hyperkrychli⁶, která je zkosená podle své hlavní diagonály. Na základě toho je znatelný rozdíl mezi původní Perlinovou šumovou funkcí a simplexovým šumem. Perlinova šumová funkce používá hyperkrychli s 2^N vrcholy a simplexový šum používá pro N dimenzí $N + 1$ vrcholů.

Simplexová šumová funkce používá přímé sčítání příspěvků každého vrcholu útvaru, který je použit na základě počtu dimenzí. V případě 2D by to byl rovnostranný trojúhelník. Bod, který je uvnitř jednoho trojúhelníku, bude počítat pouze s

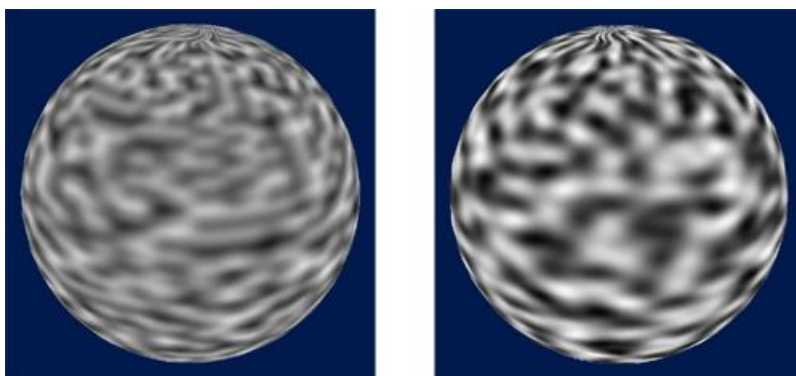
⁶ Hyperkrychle je vícerozměrná krychle.

příspěvky okolních vrcholů trojúhelníku. Ostatní vrcholy trojúhelníků nebere v úvahu, protože v nich neleží. Znázorněno na Obr. č. 19.



Obr. č. 19 Body ve 2D prostoru – výsledek se získává sečtením pouze červeně zakroužkovaných bodů.

Zdroj: [Gustavson, 2005]



Obr. č. 20 Porovnání původní Perlinovy šumové funkce (vlevo) se simplexovým šumem (vpravo) pro 2D.

Zdroj: [Gustavson, 2005]

3.1.4 Vylepšení Perlinovy šumové funkce

V roce 2002 Perlin publikoval vylepšení své původní šumové funkce [Perlin, 2002]. I když byla původní Perlinova šumová funkce efektivní, obsahovala dva nedostatky:

- Interpolační funkce $3t^2 - 2t^3$ měla druhou derivaci nenulovou v bodech $t = 0$ a $t = 1$,
- zdlouhavá metoda výpočtu derivací (gradientu).

Jak již bylo popsáno u původní Perlinovy šumové funkce, Perlin interpoloval body v mřížce (či prostoru) pomocí Hermitovy spojité funkce

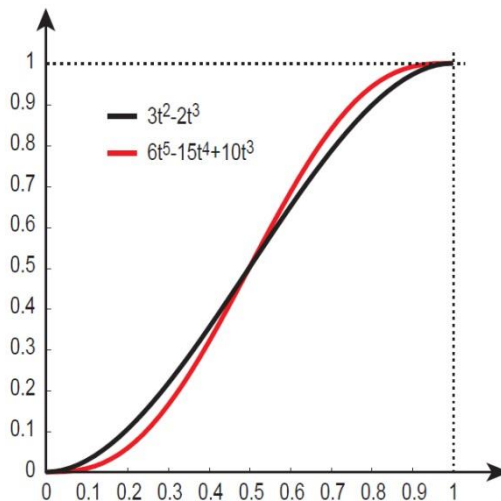
$$f(t) = 3t^2 - 2t^3,$$

kteřá má druhou derivaci rovnu $6 - 12t$, a ta není nulová ani pro $t = 0$ ani pro $t = 1$. Nenulovost druhé derivace má za následek nespojitost, která se pak následně projevuje při stínování částí povrchu, který byl pomocí Perlinovy šumové funkce vygenerován.

Aby tuto vadu opravil, nahradil Hermitovu spojitou funkci novou interpolační křivkou, polynomickeou funkcí pátého stupně

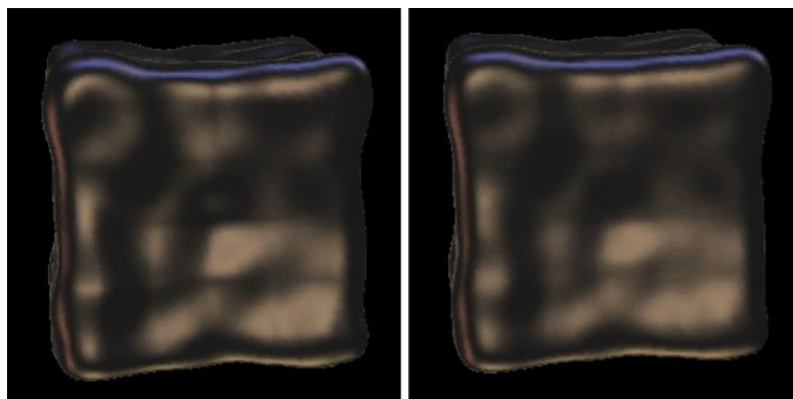
$$f(t) = 6t^5 - 15t^4 + 10t^3,$$

kteřá se původní interpolační křivce velmi podobá, ale již má nulovou druhou derivaci v obou bodech $t = 0$ a $t = 1$.



Obr. č. 21 Stará (černá) a nová (červená) interpolační křivka.

Zdroj: [Gustavson, 2005]



Obr. č. 22 Použití staré (vlevo) a nové (vpravo) interpolace.

Zdroj: [Perlin, 2002]

Druhým problémem byl zdlouhavý výpočet gradientů. Perlin výpočet zjednodušil nahrazením 256 vektorů pouhými 12 vektory, aniž by snížil náhodnost šumu, jelikož tu definuje pole náhodných permutací indexů. Tato redukce značně urychlila výpočet.

3.2 Procedurální generování textur pomocí fraktálů

Fraktální geometrie⁷ je vědní disciplína, jejíž základy formuloval Benoit B. Mandelbrot [Mandelbrot, 1983]. Na rozdíl od euklidovské geometrie, ve které se popisují objekty obyčejně rovnicemi, se ve fraktální geometrii k popisu objektů používají algoritmy, nejčastěji za využití rekurze.

Hlavní pojem, který se ve fraktální geometrii objevuje, je soběpodobnost, angl. *self-similarity*. Soběpodobné struktury lze rozložit na struktury, kdy každá je zmenšenou kopií sebe samé. Soběpodobnost má ve fraktální geometrii dva významy.

1. soběpodobnost přesná (deterministické fraktály)
2. soběpodobnost statistická (stochastické, náhodné fraktály)

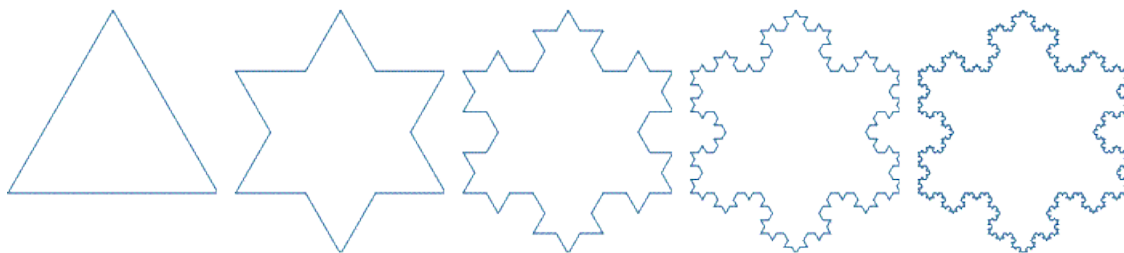
„Množina A je přesně soběpodobná, pokud je sjednocením konečného počtu transformovaných kopií sebe samé“.

„Množina B je statisticky soběpodobná, pokud je sjednocením konečného počtu zmenšených kopií sebe samé a každá z kopií má stejné statistické charakteristiky jako množina B“ [Žára et al., 2010].

Jedním z příkladů fraktálů je např. Kochova vložka od švédského matematika Helge Von Kocha, se kterou přišel roku 1904. Kochova vložka patří mezi přesně soběpodobné fraktály. Základním prvkem je trojúhelník se stranami o délce $s = 1$. P_1 je obvodem křivky (v prvním případě trojúhelníku). Každou iterací⁸ se každá strana trojúhelníku rozdělí na třetiny a v každé prostřední třetině se sestrojí nový trojúhelník o délce právě té třetiny. Poté se u každého nového trojúhelníku odstraní podstava. Výsledkem první iterace Kochovy vložky je šesticípá „Davidova hvězda“ [Nowell, 2013b]. Další iterace jsou ukázány na Obr. č. 23.

⁷ Fraktální geometrie je intenzivně rozvíjena zhruba od šedesátých let dvacátého století.

⁸ Iterace je opakování určitého vzoru (rekurze).



Obr. č. 23 Čtyři iterace Kochovy vločky.

Zdroj: [Nowell, 2013b]

Mezi nejznámější fraktály patří již zmíněná Kochova vločka, Mandelbrotova množina [Nowell, 2013a], Sierpinského trojúhelník [Conversano et al., 2011], Juli-ova množina [Tan, 1990], Cantorova množina [Soltanifar, 2006] či Mengerova houba [Menger, 1928]. V přírodě je mnoho jevů, které působí fraktálním dojmem. Příkladem takovýchto přírodních fraktálů je např. brokolice zvaná „Romanesco“ (Obr. č. 24), ulita hlemýždě (Obr. č. 25), většina rostlin, např. slunečnice (Obr. č. 26) nebo paví peří (Obr. č. 27).



Obr. č. 24 Brokolice Romanesco.

Zdroj: [McNally, 2010]



Obr. č. 25 Ulita hlemýždě.

Zdroj: [Noodle Nuke, 2015]



Obr. č. 26 Slunečnice.

Zdroj: [Cohabitaire]



Obr. č. 27 Páv a jeho peří.

Zdroj: [Cohabitaire]

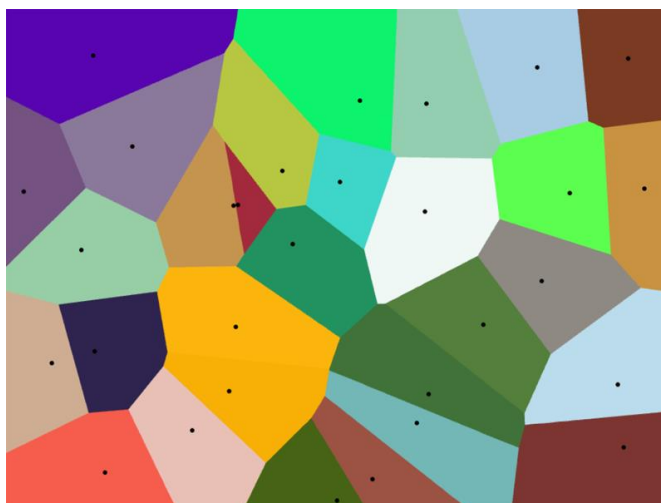
3.3 Buněčné texturování

Buněčné texturování bylo poprvé představeno Stevenem Worleyem [Worley, 1996]. Šum jako takový neumí vytvářet motiv šupin, organické tkáně, mozaiky,

kamenných oblázků, dlaždic apod. Buněčné texturování proto slouží jako výborný doplněk k šumovým funkcím, protože právě zmíněné povrchy umí napodobit.

Základem buněčného texturování je funkce n -tého nejbližšího bodu (angl. *n-th-closest basis function*), která je nazývána základní funkcí, jelikož se dá použít stejně univerzálně jako např. šum, který sám o sobě dokáže popsat pouze malé množství textur, avšak ve spojení s dalšími metodami se tato škála značně zvýší.

Tato funkce je založena na skupině bodů, které jsou náhodně distribuovány po ploše. Pro jakoukoliv polohu x existuje alespoň jeden náhodný bod takový, že žádný jiný náhodný bod neleží k bodu x blíže. Funkce $F_1(x)$ je definována jako vzdálenost x k tomuto nejbližšímu bodu. Se změnou polohy x se mění i vzdálenost x od daného bodu a tím se mění i funkce F_1 . V určité pozici bude vzdálenost bodu x ekvivalentní ke dvěma bodům zároveň, tzn. x bude mít v jednom momentu dva nejbližší body. Avšak i v této situaci bude hodnota funkce $F_1(x)$ stále jednoznačně definována, jelikož vzdálenost k oběma bodům je stejná. Ve chvíli, kdy je vzdálenost k oběma bodům stejná se vytvoří pomyslná hranice mezi buňkami a se změnou nejbližšího bodu se x již bude nacházet v nové buňce. Výsledkem je Voronoiův diagram [Aurenhammer et al., 2000] ukázán na Obr. č. 28.

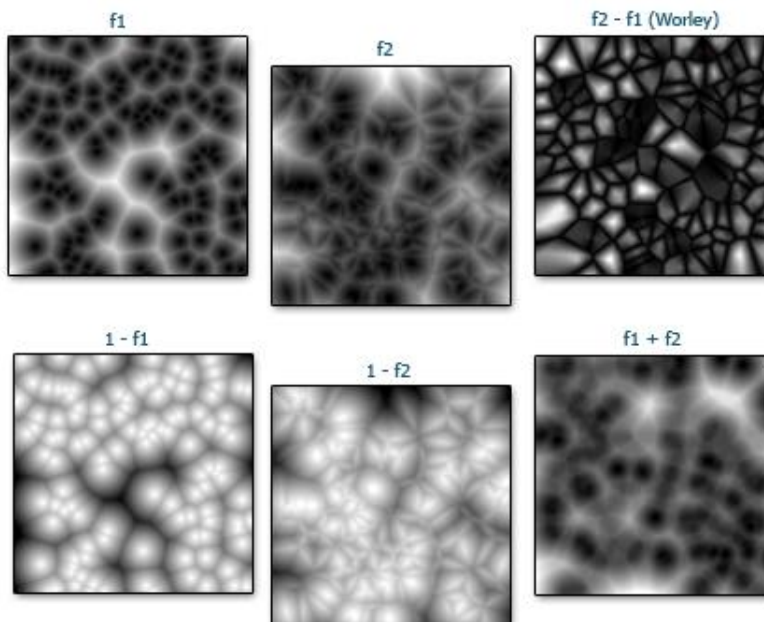


Obr. č. 28 Voronoiův diagram.

Zdroj: [Beutel, 2015]

Funkce F_2 může být definována jako vzdálenost x k druhému nejbližšímu bodu a stejně tak může být definována funkce F_n jako vzdálenost x k n -tému nejbližšímu bodu. Na Obr. č. 29 funkce F_1 představuje vzdálenost k prvnímu nejbliž-

šimu bodu a F_2 představuje vzdálenost k druhému nejbližšímu bodu. Dále jsou ukázány inverze k oběma funkcím $1-F_1$ a $1-F_2$, součet funkcí $F_1 + F_2$ a rozdíl funkcí $F_2 - F_1$.



Obr. č. 29 Vizuální podoba funkcí F_1 , F_2 a jejich variant.

Zdroj: [McCombs]

Nejjednodušší texturou na základě buněčného texturování by bylo mapování barvy pouze nízkým hodnotám funkce F_1 . Toto by na textuře vytvořilo efekt „polka dot“.

3.4 Procedurální generování pomocí Lindenmayerových systémů

Lindenmayerovy systémy (zkráceně L-systémy) jsou skupinou fraktálů definovaných ve své nejjednodušší podobě za pomoci přepisovacích gramatik. Základem je přepisování základního řetězce podle stanovených pravidel. Ke každému symbolu v řetězci se přiřadí určitý geometrický význam, například nějaká transformace, generování objektu, či akce [Žára et al., 2010].

Veškeré vlastnosti L-systémů, podobně jako u původních fraktálů, jsou založeny na rekurzi. Pomocí L-systémů lze generovat fraktální objekty, které vypadají jako rostliny, stromy a další přírodní fenomény.

L-systémy lze z pohledu míry náhodnosti rozdělit na:

- *Deterministické*
- *Stochastické – náhodné*

Deterministický bezkontextový L-systém je tvořen uspořádanou trojicí:

$$G = [V, P, S]$$

kde:

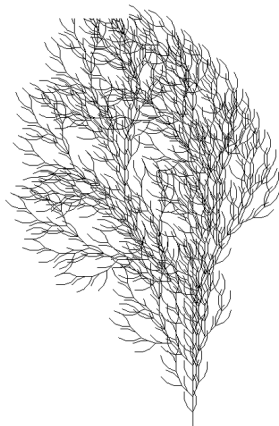
- *V je konečná abeceda symbolů,*
- *P je konečná množina pravidel tvaru $A \rightarrow B$; $A \in V$; $B \in V^*$,*
- *S je axiom (tzv. semínko) – neprázdná posloupnost symbolů $S \in V^+$. Jde o počáteční stav konkrétního L-systému.*

Posloupnosti symbolů z abecedy se nazývají slova nebo řetězce. Množina všech slov nad abecedou V je V^* . Množina všech neprázdných slov nad abecedou V je V^+ .

Podstatu determinismu určuje podmínka, že množina P nesmí obsahovat dvě pravidla se stejnou levou stranou. Při přepisování je díky této podmínce vždy jasné, které pravidlo pro přepis vybrat [Tišnovský, 2006a].

Na Obr. č. 30 je ukázka deterministického L-systému, který má následující předpis:

- Axiom = F
- $F \rightarrow FF+[+F-F-F]-[-F+F+F]$
- Úhel = 22,5°



Obr. č. 30 Keř vygenerovaný pomocí L-systémů.

Zdroj: [Bourke]

Deterministické L-systémy podle určitého pravidla (či pravidel) budou po vygenerování vypadat vždy stejně. Náhodnost je tedy v jejich případě nulová. Pokud by bylo cílem vnést do simulované rostliny jistou náhodnost, tedy aby po vygenerování vypadala pokaždé trochu jinak, lze použít stochastické L-systémy.

Stochastické L-systémy se od těch deterministických liší tím, že množina P již smí obsahovat dvě pravidla se stejnou levou stranou [Tišnovský, 2006b]. Těmto pravidlům se jednotlivě přiřadí míra pravděpodobnosti, která se zpravidla píše před nebo za pravidlo do kulatých závorek „(0.5)“.

Příklad:

$$F \rightarrow F[+F]F[-F](0.3)$$

$$F \rightarrow F[+F]F[+F](0.7)$$

Tento příklad demonstruje pravidla se stejnou levou stranou, tedy s písmenem „F“, nicméně pravé strany se trochu liší. Na míře odlišnosti nezáleží, mohou se lišit diametrálně nebo jen trochu. V závorkách jsou uvedeny pravděpodobnosti, se kterou budou daná pravidla při výpočtu vybírána. Dohromady by součet pravděpodobností měl být roven jedné. Ve výše uvedeném příkladu bude druhé pravidlo vybíráno častěji, protože má větší míru pravděpodobnosti pro výběr.

Náhodnost stochastických L-systémů tedy spočívá ve výběru mezi několika pravidly se stejnou levou stranou s určitou mírou pravděpodobnosti vybrání. Čím více bude takových pravidel, tím náhodnější budou výsledné výstupy.

L-systémy jsou v současné době nejlepší propracovanou formální teorií pro modelování syntetických rostlin. Výsledné obrazce působí velmi realisticky. Tato práce je zaměřena na 2D procedurální textury, pomocí L-systémů se ale běžně generují i 3D modely. Na Obr. č. 31 je ukázka velmi působivých a realisticky vypadajících modelů.



Obr. č. 31 Deterministické 3D rostliny vygenerované pomocí L-systémů.

Zdroj: [Allen, 2007]

3.4.1 Želví grafika

Jde o základní nástroj pro vytváření deterministických L-systémů. Želví grafika (angl. turtle graphics) geometricky interpretuje L-systém. Hlavním aktérem v tomto nástroji je želva, která reprezentuje zařízení, které vytváří obrazce složené z úseček. Želva je definována svým stavem a tabulkou akcí, jež může provádět. Stav želvy obsahuje polohu želvy (kde se nachází) a orientaci želvy (kam je želva natočena). Želva se řídí sekvenčně zadaným řetězcem a jednotlivé symboly jsou pro ni příkazy, které má vykonat [Tišnovský, 2006a].

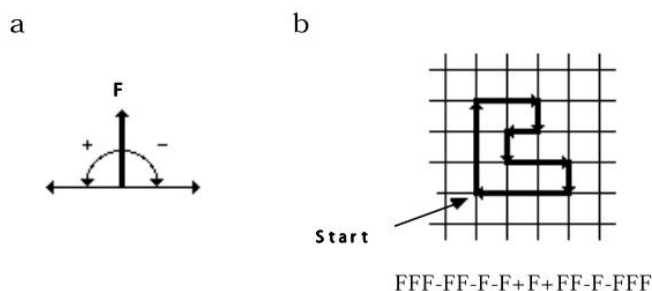
Základní symboly, které želví grafika může obsahovat:

Symbol	Příkaz	Význam
F	draw forward	želva se posune dopředu a nakreslí úsečku
G	move forward	želva se posune dopředu bez nakreslení úsečky
B	draw backward	želva se posune dozadu a nakreslí úsečku
+	turn left	želva se natočí doleva o předem známý počet stupňů
-	turn right	želva se natočí doprava o předem známý počet stupňů

Tabulka 1 Základní symboly želví grafiky.

Zdroj: [Tišnovský, 2006a]

Díky těmto symbolům se může želva posouvat dopředu nebo dozadu bez kreslení čáry i s kreslením čáry (tj. úsečka nakreslená mezi počáteční pozicí a koncovou pozicí želvy) a otáčet se okolo své osy doprava a doleva. Dají se tak vytvářet různé geometrické tvary, které ale nemohou být větvené, to znamená, že želva se po nakreslení čáry nemůže vrátit o krok zpět a vydat se jiným směrem, aby tak nasimulovala rozvětvení rostliny, což je základem u simulace rostlin.



Obr. č. 32 (a) Želví interpretace symbolů F , $+$, $-$. (b) Interpretace pohybu želvy podle symbolů. Úhel otočení želvy je roven 90° . Na počátku želva směřuje nahoru.

Zdroj: [Prusinkiewicz, 2004]

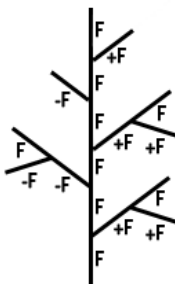
3.4.2 Závorkové L-systémy

K vytvoření rostlin pomocí L-Systémů nelze použít Deterministické L-systémy nultého řádu (D0L-systémy). D0L-systémy jsou nejjednodušší formou L-systémů, ale postrádají možnost větvení struktury [Prusinkiewicz, 2004]. Právě díky větvení je možné vytvářet rostliny, tj. traviny, keře, stromy a další. Závorkové L-systémy tento problém odstraňují, neboť struktury L-systémů větvit umí, jsou proto vhodné k vytváření rostlin.

Větvená struktura rozlišuje začátek a konec vedlejší větve, k čemuž slouží symboly pravé a levé hranaté závorky „ $[]$ “. Do závorek se vkládají samostatné sekvence příkazů a po vykonání řetězce uloženého mezi $[]$ se želva vrátí do předchozí (původní) pozice. Působí to jako skoky do podprogramu a návratu z něj [Tišnovský, 2006b].

Na Obr. č. 33 je vidět, jak závorkové L-systémy fungují. Předpis pro L-systém na obrázku má podobu:

$F[+F[+F]F]F[-F[-F]F]F[+F[+F]F]F[-F]F[+F]F$



Obr. č. 33 Graficky znázorněný závorkový L-systém s pravidly.

4 Testování metod a hodnocení vybraných aplikací

V předchozí kapitole byly jednotlivé metody pro procedurální generování rozebrány teoreticky. Tato kapitola představuje existující aplikace, ve kterých jsou jednotlivé metody testovány. Kritérii pro výběr aplikací byla schopnost práce s analyzovanými metodami, kvalita generovaného výstupu, možnosti nastavení a intuitivnost ovládání.

Aplikace jsou rozděleny do sekcí podle metod, na základě kterých generují výstupy. Popis aplikací zahrnuje jejich základní funkce, omezení, dostupnost, přehlednost a složitost ovládání. Každá aplikace je zhodnocena a obsahuje připojenou ukázkou vygenerovaných textur.

4.1 Perlinova šumová funkce

Perlinova šumová funkce sama o sobě se nejčastěji používá pro generování textur dřeva, mramoru, oblak či kouře. Následující aplikace byly vybrány, neboť umí generovat textury právě pomocí této funkce.

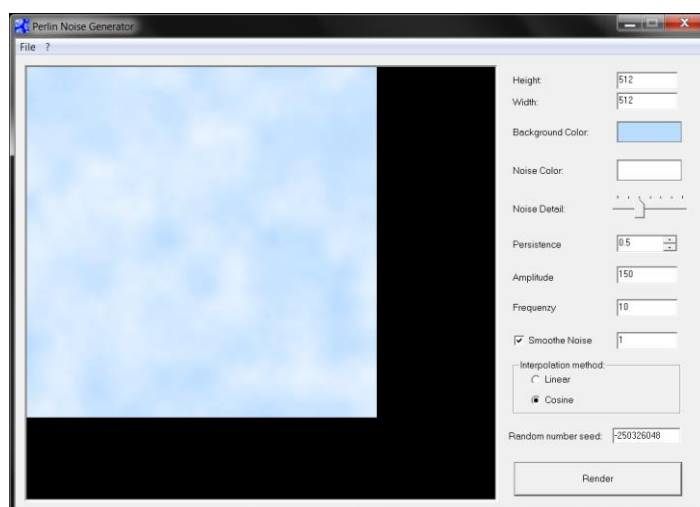
4.1.1 Perlin Noise Generator

Tento software je volně ke stažení a byl vyvinut na internetovém serveru Runicsoft [Hartl, 2011]. Nevyžaduje instalaci. Perlin Noise Generator umí vygenerovat pouze základní Perlinovu funkci s nastavením těch nejnужnějších parametrů. Nelze od něj očekávat složité a zvláště kvalitní textury, ale na něco jednoduchého postačí.

Neumí generovat bezešvé textury. Maximální možná velikost vygenerované textury je 512×512 px (pixelů). V aplikaci je možné nastavit barvu šumu i barvu pozadí, detail a jemnost šumu, persistenci, amplitudu, frekvenci a interpolační metodu (lineární nebo kosinovou). Výslednou texturu lze uložit ve formátu BMP (24bit), v PNG nebo ve speciálním formátu NF (Noise File), který zkládá nastavení parametrů pro generování. Lze načíst pouze soubor s koncovkou „.nf“. Generátor renderuje až po zmáčknutí tlačítka Render.

Demonstrační textura oblak (viz Obr. č. 34) byla nastavena na maximální velikost, tedy 512×512 px, ostatní parametry byly ponechány na původních hodno-

tách (persistence = 0,5; amplituda = 150; frekvence = 10; jemnost šumu = 1; interpolační metoda = kosinová).



Obr. č. 34 Náhled aplikace Perlin Noise Generator.

4.1.2 TextureGenerator-Online

Jedná se o volně dostupný na webu umístěný generátor [Petry], jak již název napovídá. Kromě Perlinovy šumové funkce umí vygenerovat i složitější textury, například cihly, terén, dlaždice, či abstraktní vzory. Software renderuje automaticky.

U Perlinovy šumové funkce lze nastavit barvu funkce a barvu pozadí, počet oktáv, měřítko (úroveň přiblížení/oddálení), persistence, semínko pro pseudonáhodný generátor (*seed*) a rotace. Rotace nefunguje tak, jak by měla, tedy po otočení se vygeneruje pouze otočená textura s viditelným přechodem, což je nežádoucí. Pomocí semínka pseudonáhodného generátoru je možné zajistit vygenerování po každé jiné textury při následných spuštěních programu.



Obr. č. 35 Náhled aplikace TextureGenerator-Online.

Zdroj: [Petry]

Výslednou texturu lze uložit ve formátu PNG či JPEG v pěti různých velikostech od 256×256 px až po 4096×4096 px. Tento software bohužel neumí aplikovat na Perlinovu funkci turbulenci, díky které lze vytvořit např. texturu dřeva nebo mramoru. Stejně tak jako Perlin Noise Generator je i tento generátor jednoduchý na ovládání, ale chybí mu pokročilé možnosti nastavení.

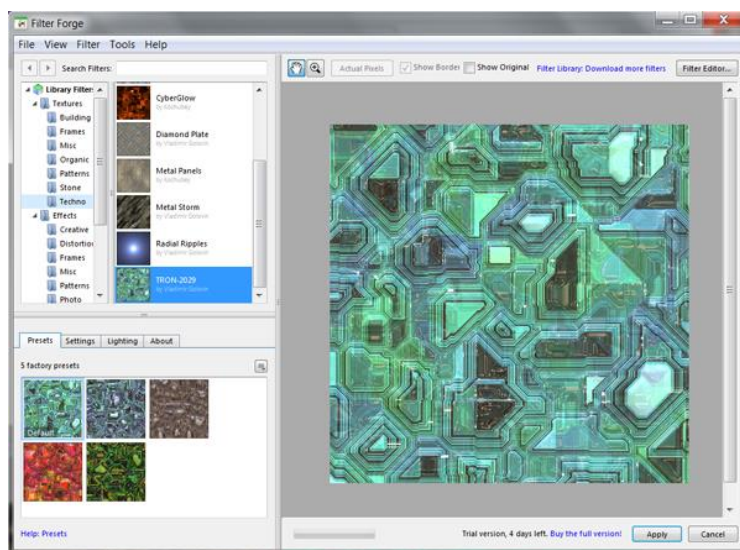
4.1.3 Filter Forge

Filter Forge je plug-in⁹ pro Adobe Photoshop, který umožňuje vytváření vlastních filtrů (předpisů pro textury). Tento plug-in ale není zdarma a tak byla pro účely práce využita trial verze Filter Forge, konkrétně verze 5.0 na 30 dní, dostupná z [Filter Forge, Inc., 2016]. Verze je pouze časově omezena, nejsou tedy omezeny žádné funkce. Plug-in vyžaduje instalaci.

Ve Photoshopu je Filter Forge k nalezení pod záložkou „Filtr“ (angl. Filters). Před spuštěním Filter Forge musí mít uživatel na paměti, že pokud má ve Photoshopu aktivní okno s obrázkem ve velikosti např. 1920×1080 px, výsledná textura bude mít stejné rozlišení. Z toho plyne, že jestliže je požadována textura ve velikosti např. 512×512 px, musí být otevřen obrázek v tomto rozlišení, což může být matoucí z hlediska uživatelské přívětivosti. Textura se vždy vytvoří jako nový ob-

⁹ Jde o software, který nepracuje samostatně, ale jako doplňkový modul jiné aplikace a rozšiřuje tak její funkčnost.

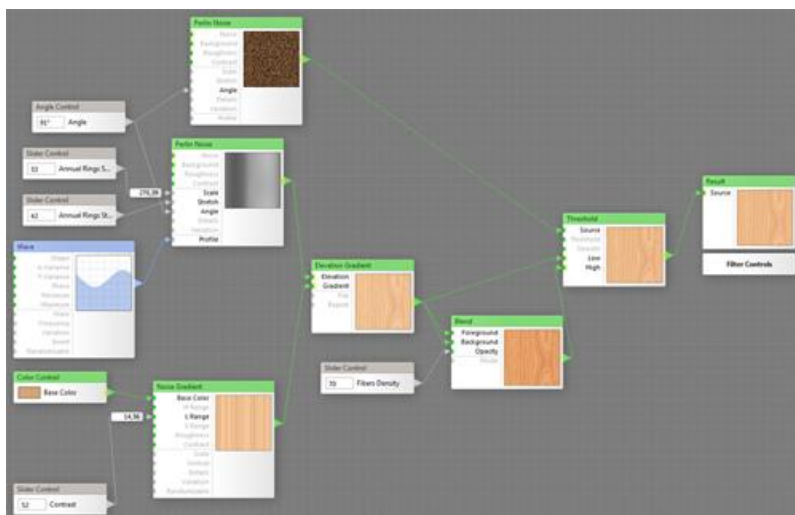
rázek, nikdy nepřepíše žádný, který byl původně otevřený, nezáleží tedy na obsahu aktivního okna obrázku.



Obr. č. 36 Náhled plug-inu Filter Forge.

Filter Forge má k dispozici celou řadu komponent, ze kterých si uživatel může vybírat při popisování a vytváření textur. Na první pohled se může zdát velmi složitý, ale složitost je způsobena především velmi rozsáhlými možnostmi nastavení, jinak je uživatelsky přívětivý a dobře se s ním pracuje. Výsledné textury jsou vizuálně velmi kvalitní. Renderuje automaticky.

Jako první příklad použití Filter Forge byl zvolen základ textury dřeva a ten byl dále upravován. Textura se skládá z mnoha prvků, které ji popisují, ať už je to funkce, efekt či obyčejná barva. Perlinova šumová funkce je zde použita dvakrát. Jednak vytváří efekt šumu, roztaženého vertikálně, který zdrsňuje povrch dřeva a druhé použití Perlinovy funkce zajišťuje úroveň detailů na výsledné textuře.



Obr. č. 37 Ukázka prvků, ze kterých se skládá výsledná textura.

Jednotlivé prvky se propojují tažením myši ze zelené šipky umístěné vpravo na každém okně s efektem, funkcí, nebo hodnotou. Poslední okno vpravo je výsledná textura.

Cílem demonstrace plug-inu bylo vytvořit textury, které by napodobily bukové dřevo a bílý mramor.

Bukové dřevo se vyznačuje světlou krémovou až růžovo-hnědou barvou. Jeho vlákna jsou rovná a většinou širšího charakteru [Dřevo Spirit of Angmar & Bithynie]. Cílem bylo nasimulovat tyto vlastnosti pomocí procedurální textury.



Obr. č. 38 Předloha bukového dřeva.

Zdroj: [Rockpanel Group]



Obr. č. 39 Výsledná textura bukového dřeva vytvořeného přes Filter Forge.

Pro texturu mramoru byl zvolen typ zvaný „Arabescato“. Na tuto texturu byla použita komponenta zvaná „Noise Distortion“, která představuje turbulenci. Hodnota turbulence byla nastavena na 25 (maximální hodnota je 100, v teoretické

části této práce byla turbulence uváděna v rozmezí 0-1, takže by tato hodnota odpovídala 0,25).



Obr. č. 40 Předloha mramoru.

Zdroj: [All Granite and Marble Corp., 2015]



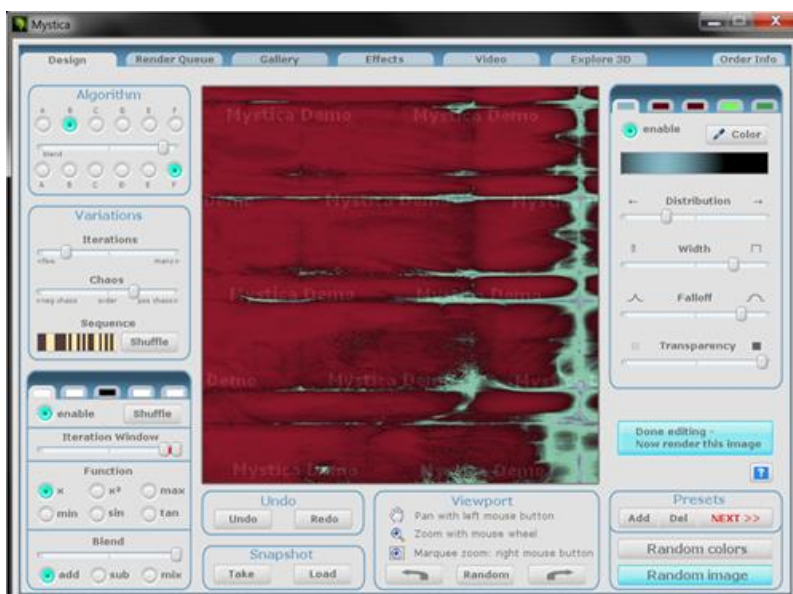
Obr. č. 41 Výsledná textura mramoru vytvořená přes Filter Forge.

4.2 Fraktály

Pomocí fraktálu je možné vytvářet zajímavé tvary a obrazce. Následující aplikace jsou schopné tyto jevy generovat. Následující podkapitoly popisují vlastnosti jednotlivých aplikací a pro každou z nich bude předveden výsledný vygenerovaný fraktál.

4.2.1 Mystica

Tento nástroj 2D a 3D fraktálů není zdarma a proto pro účely této práce byla použita časově omezená verze na 30 dní, která je dostupná z [Dawntec]. Na výsledných texturách s fraktály se objevuje ochranný nápis „Mystica Demo“. Mystica vyžaduje instalaci. V této práci byla použita verze 5.7.



Obr. č. 42 Náhled aplikace Mystica.

Mystica je určena pro uživatele, kteří nechtějí nebo neumí zadávat celý předpis pro fraktál, především pro nenáročné uživatele, kteří nepotřebují mít na prostou kontrolu nad podobou fraktálu. Software nabízí 6 předpřipravených funkcí a těmi jsou: x , x^2 , maximum, minimum, sinus a tangens. Fraktál se generuje náhodně a těmito funkcemi ho lze pouze usměrnit, nikoliv zadat tvar. Mystica při každém přepnutí okna (či prvním zapnutí softwaru) sama generuje náhodný fraktál, který uživatel může pouze ovlivnit.

Na výběr je 6 algoritmů, které fraktál mírně pozměňují. Dále lze upravovat posuvníky pro množství iterací a míru chaosu. Iterace v tomto programu nejdu zadávat číselně, posuvník má pouze levou stranu jako „málo“ a pravou jako „hodně“ iterací. Taktéž je na tom i posuvník chaos, který má levou stranu „negativní chaos“ a pravou „pozitivní chaos“. Pomocí posuvníků jsou řešeny i všechny ostatní parametry, kterými jsou rozložení barevného spektra, šířka (myšleno pro barevné spektrum), rychlost klesání/stoupání intenzity od středu spektra a průhlednost.

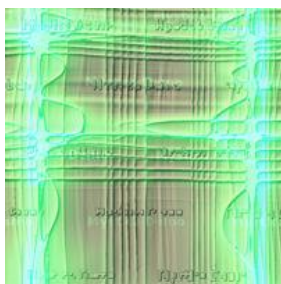
Na přepínání mezi náhodnými fraktály slouží i tlačítko „Random image“ a náhodné barvy lze generovat tlačítkem „Random colors“. V okně lze s fraktálem posouvat (levým tlačítkem myši), přibližovat/oddalovat (kolečkem myši) a vybrat areu pro maximální okamžité přiblížení (pravé tlačítko myši). Lze ho také otáčet o 90° doprava i doleva.

Upravené fraktály v první záložce Design se ukládají do záložky Render Queue, kde je lze nechat vyrenderovat (v rozlišení od 256×256 až do 8000×6000 px, 16 různých rozlišení) a v záložce Gallery následně kopírovat, editovat, otevřít v prohlížeči fotografií (Windows), smazat, upravit efekty, vytvořit video (což není náplní tohoto popisu), exportovat obraz ve formátech JPEG, PNG, BMP, TIF, TGA a MYS.

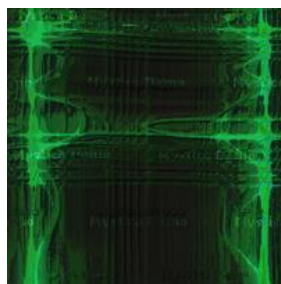
Samotné základní nastavení je zadané a nyní se fraktál může dále upravovat v záložce Effects. Na fraktál lze aplikovat různé filtry pro zaostření, rozostření, zvýraznění hran, reliéf a další. Zajímavou funkcí je 3D Light (3D světlo), která po zapnutí vytvoří na textuře 3D světelný efekt, kde se nejsvětlejší body v obraze jakoby posunou po z-ové ose dopředu. Tomuto efektu se nastavuje pozice světla, způsob odrazu (jestli spekulární nebo rozptýlené světlo) a míra prolnutí. Také jsou k dispozici tři klasické úpravy: jas, kontrast a gamma.

Patrně nejzajímavější v této záložce je poslední box úprav a to vytváření vrstev. Lze vytvořit vrstvu pouze kopírováním té původní, ale i tak se tím fraktály mohou ozvláštnit. Nové vrstvě lze upravit šířku, jas, změnit metodu prolnutí a míru průhlednosti. Vrstvy lze vypnout a vrátit se tak k původní textuře. Upravovat hodnoty vrstev lze vždy pouze na poslední vrstvě. Nelze mezi nimi přepínat.

Následující verze textur ukazují působení efektů na původní texturu.



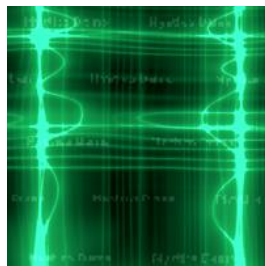
Obr. č. 43 Verze A textury s efekty.



Obr. č. 44 Verze B textury s efekty.



Obr. č. 45 Verze C textury s efekty.



Obr. č. 46 Původně vygenerovaná textura bez efektů.

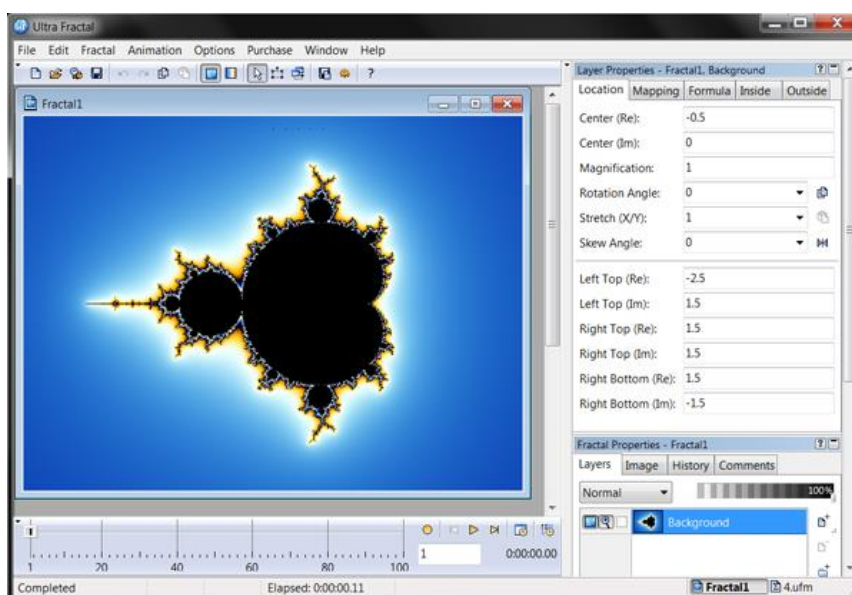
Pro verzi A byl aplikován filter „soften“ a bylo použito 3D světlo s mírně sníženou pozicí světla a odrazem nastaveným téměř na 100 % „specular“. Na texturu bylo také použito vrstvení s metodou prolnutí „add“. Verze B se od verze A liší změnou metody prolnutí vrstev z „add“ na „and“. Pro třetí verzi C byl změněn filter na „edges 2“, byl vypnut efekt 3D světla, byl zvýšen jas, o něco méně i kontrast a metoda prolnutí pro vrstvy byla vrácena na metodu „add“.

Na původní texturu při jejím vytváření v záložce Design bylo použito toto nastavení: Algoritmus E, posuvník pro iterace se nacházel na polovině „few“, chaos byl posunut mírně doleva k „neg chaos“. Byla vybrána funkce tangens.

4.2.2 Ultra Fractal

Ultra Fractal je pokročilý nástroj pro generování fraktálů. Tento software není zdarma a tak pro účely této práce byla použita časově omezená verze na 30 dní. Na výsledných texturách s fraktály je proto ochranný nápis „Evaluation copy“. Demo aplikace je dostupné z [Slijkerman, 2010] a vyžaduje instalaci. Pro tuto práci byla použita verze 5.04.

Tento generátor je velmi složitý a velmi schopný rychle generovat různé fraktály tím, že poskytuje vlastní rovnice (algoritmy) pro generování fraktálu. Má ale i předem vytvořené některé ze základních fraktálů jako třeba Mandelbrotova množina, Sierpinského trojúhelník a mnoho dalších.



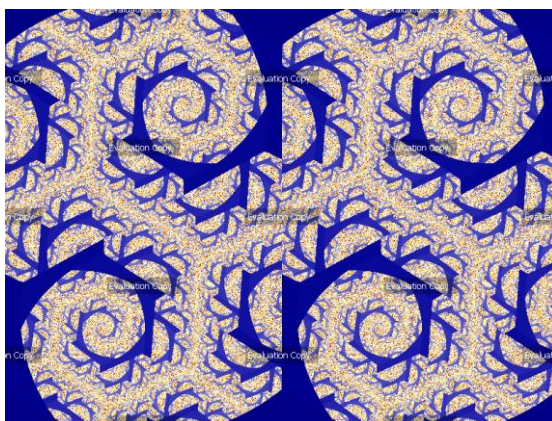
Obr. č. 47 Náhled aplikace Ultra Fractal s Mandelbrotovou množinou.

Vygenerované fraktály umí uložit ve formátech BMP, JPEG, PSD, PNG, TGA a TIFF. Projekt s veškerým nastavením ukládá ve formátu UFR (Fractal file). Ultra Fractal neumí jen generovat fraktály, umí i vytvářet animace, jejichž popis je nad rámen této práce.

Jak již bylo zmíněno výše, Ultra Fractal umožňuje vytváření vlastních předpisů pro zadání tvaru a následného generování fraktálu. Jako testovací fraktál k následnému upravování a zkoumání byl vybrán následující předpis:

```
barnsleyj1 {
init:
  z = pixel
loop:
  IF real(z) >= 0
    z = (z - 1) * @start
  ELSE
    z = (z + 1) * @start
  ENDIF
bailout:
  |z| < 4
default:
  title = "Barnsley 1 (Julia)"
  maxiter = 149
  param start
    caption = "Parameter"
    default = (0.6, 1.1)
  endparam
}
```

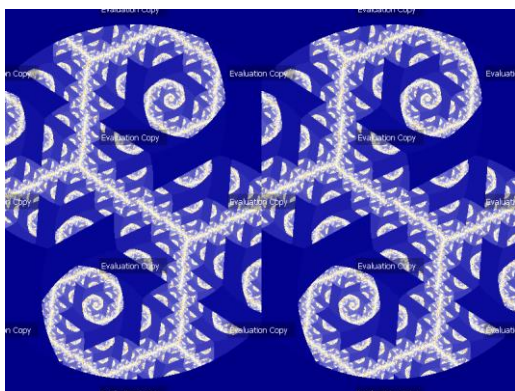
Původní verze tohoto předpisu je znázorněna na Obr. č. 48. Změny výsledného fraktálu po úpravě parametrů předpisu jsou patrné na Obr. č. 49 a Obr. č. 50.



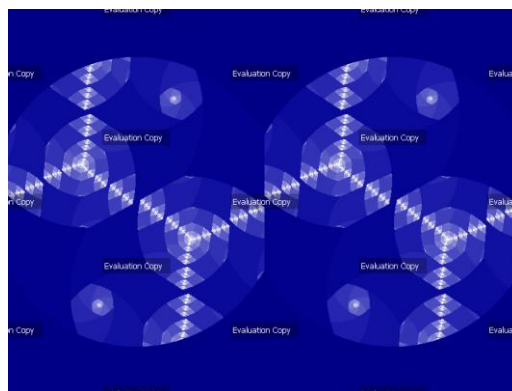
Obr. č. 48 Barnsley (základem je Juliova množina).

Předpis fraktálu je realizován pomocí vestavěného programovacího jazyka. Ze všeho nejdříve se do klauzule „init“ deklarují proměnné, se kterými má být

v kódu operováno. Do samotné smyčky, tedy „loop“, se zapisuje tvar fraktálu, ke kterému se používají zadané proměnné a různé funkce. Do smyčky lze používat i podmínky, v této ukázce byla použita podmínka „if“. „Bailout“ hodnota specifikuje rozsah funkce, která určuje podmínku pro ukončení iterace. Sekce „default“ deklaruje základní hodnoty jako např. název fraktálu, který se poté zobrazuje při výběru předpisu, nebo maximální počet iterací. Parametr v sekci „param“ s názvem „start“ deklaruje výchozí hodnotu funkce. Použitím hodnot 0,6 a 1,1 byla tato výchozí proměnná automaticky nastavena na datový typ float.

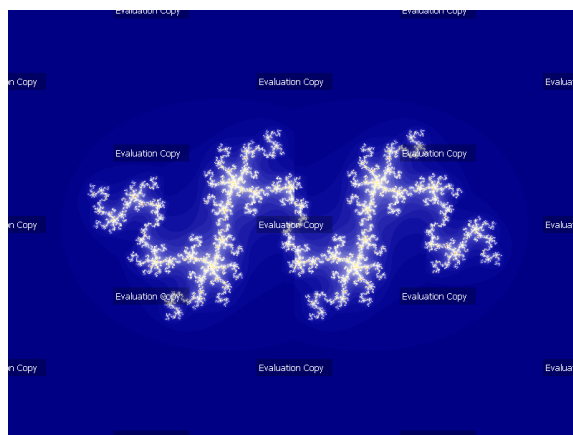


Obr. č. 49 První změna původního předpisu.



Obr. č. 50 Druhá změna původního předpisu.

Pro získání fraktálů na Obr. č. 49 a Obr. č. 50 byla změněna hodnota „bailout“ na hodnotu 3 (vlevo) a 2 (vpravo). Pro větší změnu výsledné podoby fraktálu byla změněna funkce na $z = (z - 1) * z * @start$, stejně tak v klauzuli ELSE. Hodnota „bailout“ byla nastavena na $|z| < 4$. A poslední úpravou byla změna výchozí hodnoty funkce na (1.6, 1.1). Výsledný fraktál je zobrazen na Obr. č. 51.



Obr. č. 51 Výsledný fraktál po změnách.

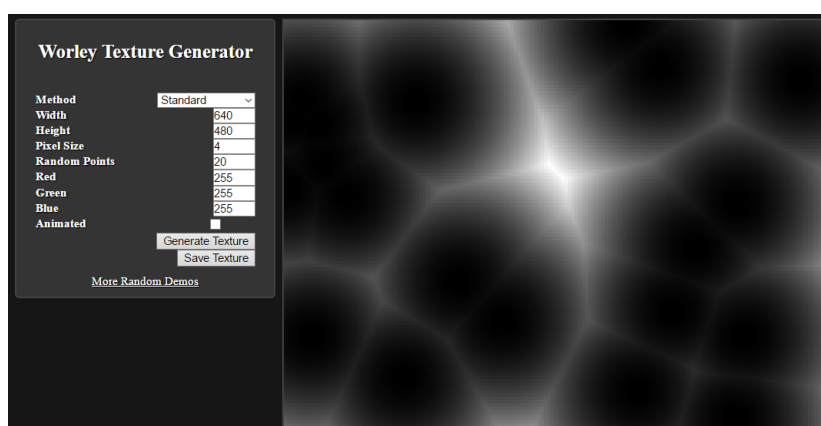
Tento software dokáže poměrně rychle vygenerovat jakkoli složité fraktály. Je poměrně složitý na pochopení, a tak chvíli trvá, než se uživatel seznámí se všemi ovládacími prvky a způsobem zápisu předpisů pro fraktály. Je určen hlavně pro náročného uživatele, kteří ovládají základy programování a matematiku. Díky vlastnímu programování má tento generátor možností mnohem více, než bylo popsáno výše. Více informací lze nalézt v zabudovaném, detailně zpracovaném manuálu, do kterého může uživatel kdykoli nahlédnout.

4.3 Buněčné texturování

V této kapitole budou předvedeny aplikace pro generování textur za pomoci buněčného texturování. Touto metodou je možné vytvořit textury podobné např. kamenům, mikroorganismům, mozaikám a dalším.

4.3.1 Worley Texture Generator

Tento generátor je volně dostupný na webu [Somethinghitme, 2010]. Jde o poměrně jednoduše ovladatelný generátor, který umí některé základní buněčné funkce, jako jsou funkce prvního nejbližšího bodu, inverzi k ní, $F_2 - F_1$, $F_1 * F_2$, a pak nové funkce $\sqrt{F_1 * F_2}$ a $\sqrt{F_2 - F_1}$. Velikost vygenerované textury je nastavitelná. Dalším nastavením je velikost pixelu, což představuje rozlišení dané textury bez změny velikosti textury. Nastavit se zde dá i počet bodů, které se mají roztribuovat po obraze, přičemž nulu tento program ignoruje.

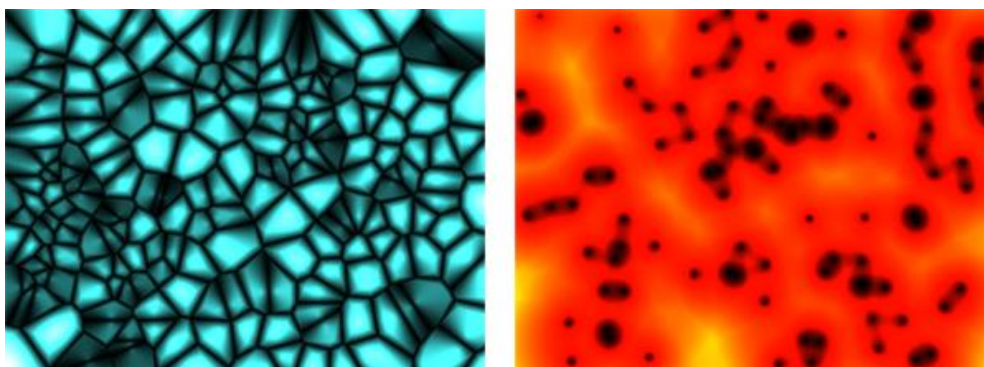


Obr. č. 52 Náhled aplikace Worley Texture Generator.

Nastavitelná je i RGB barva, jejíž hodnoty však neodpovídají standardnímu nastavení. Nelze nastavit běžné hodnoty od 0 – 255, ale je nutné zadávat různé hodnoty, které se mohou dostat i do statisíců, stejně tak mohou být i záporné. Hodnoty barev ovlivňují i poloměr černé barvy kolem zdrojového bodu, který bude vždy vykreslen a vždy bude černý. Čím větší je hodnota barvy, tím menší je černý poloměr okolo zdrojového bodu. Nastavení barev proto vyžaduje velké množství experimentování.

Software nerenderuje textury automaticky, ale stisknutím tlačítka „Generate Texture“. Po každém stisknutí tohoto tlačítka aplikace vygeneruje vždy jinou variaci textury. Umožňuje zobrazit texturu i s animací, ale animaci jako takovou neukládá. Tlačítkem „Save Texture“ zobrazí texturu v nové záložce prohlížeče, kterou si může uživatel uložit ve formátu PNG.

Ukázka textur vygenerovaných tímto programem je na Obr. č. 53.



Obr. č. 53 Výsledné ukázkové textury A (vlevo) a B (vpravo).

Nastavení textury A	Nastavení textury B
Method: D2 - D1	Method: D1 * D2
Width: 640 px	Width: 640 px
Height: 480 px	Height: 480 px
Pixel Size: 1	Pixel Size: 1
Random Points: 200	Random Points: 100
Red: 255	Red: 5000
Green: 900	Green: 255
Blue: 900	Blue: 1

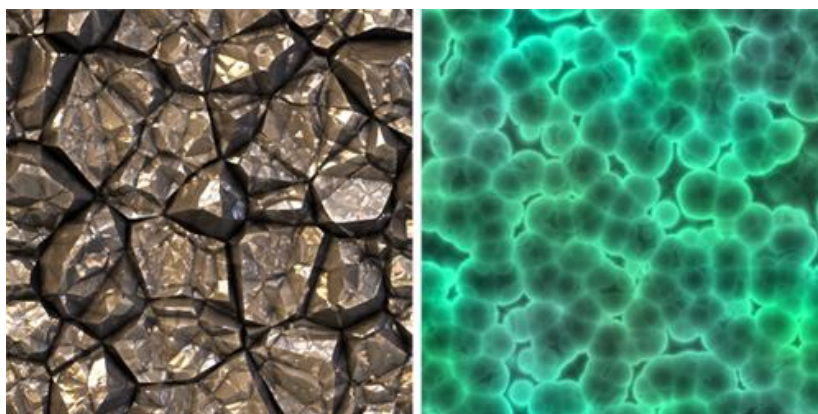
Tabulka 2 Nastavení výsledných textur vygenerovaných aplikací Worley Texture Generator.

4.3.2 Filter Forge

Filter Forge již byl představen u aplikací Perlinovy šumové funkce, která ale není jediná, kterou umí generovat. V této kapitole jsou popsány možnosti generování buněčných textur.

Filter Forge je plug-in pro Adobe Photoshop, který je schopný vytvářet dobře vypadající a realistické textury. Není ale zdarma, takže pro tuto práci je používána časově limitovaná 30 denní verze.

Pro ukázkou, jak může vypadat textura vygenerovaná pomocí buněčného texturování, byla vybrána textura kamene a mikroskopických bakterií. Obě dvě textury jsou bezešvé a ve velikosti 512×512 px. Kromě buněčného texturování byla na obě textury použita i Perlinova šumová funkce, která zde vytváří efekt zašpinění textury šumem, pro realističtější dojem.



Obr. č. 54 Výsledné textury kamene (vlevo) a mikroskopických bakterií (vpravo).

Na základě ukávek výsledných textur je patrné, že je Filter Forge výborný nástroj v generování procedurálních textur. Jeho výstupy jsou kvalitní a vypadají velmi profesionálně a realisticky.

4.4 L-systémy

Lindenmayerovy systémy, zkráceně L-systémy slouží v první řadě pro generování rostlin. V následujících aplikacích jsou simulovány různé druhy rostlin pomocí přepisovacích gramatik a následně je každá aplikace ohodnocena podle výsledků generování, srozumitelnosti, počtu funkcí a celkově uživatelského dojmu. Předveden je vždy i zápis L-systému, který se u těchto systémů zpravidla liší.

4.4.1 Lindenmayer System Generator

Jedná se o volně dostupný online generátor, který umí generovat pouze deterministické L-systémy. Dostupný z [Noland]. Je uživatelsky přívětivý a velmi jednoduchý na ovládání. Nerenderuje automaticky, až po aktivaci tlačítka Render.

V hlavní nabídce je možné si v canvasu¹⁰ kliknutím zvolit, odkud se začne daná rostlina generovat. Užitečnost této funkce je omezená, protože canvas nemá možnost scrollování¹¹, takže pokud je rostlina větší než okno prohlížeče, nebude vidět celá.

Druhý parametr nastavuje délku čáry, ze které se celý model skládá. Možné hodnoty jsou 1-8, přičemž aplikace neuvádí přesný přepočtení těchto hodnot na počet pixelů. Třetí parametr nastavuje počet iterací, přičemž nejnižší hodnota je 3, což může být nepříjemné, pokud cílem je vygenerovat pouze jednoduchý fraktál s jednou nebo dvěma iteracemi. Maximální počet iterací je 10, což může představovat problém naopak při generování složitých fraktálů.

Nastavení úhlů je již uživatelsky plně nastavitelné zadáním vlastního čísla. Úhly mohou být i záporné. Nastavuje se počáteční úhel, do jakého se začne model generovat a dále se nastavuje úhel při otočení.

Generátor umožňuje nastavit počáteční písmeno, tedy axiom. Ten je potřebný pro další možnosti nastavení a těmi jsou přepisovací pravidla. Jak již bylo zmíněno výše, tento generátor umí pouze deterministické L-systémy, tudíž vždy jen pravidlo s unikátní levou stranou. Přidaná pravidla jdou upravit či smazat. Pro základní L-systémy toto nastavení stačí.

Na Obr. č. 55 je ukázka vygenerované rostliny. Detail rostliny působí nepřírovně z důvodu deterministického přístupu ke generování. Nicméně i tak lze v omezené míře dosáhnout toho, aby celkový dojem z rostliny nepůsobil příliš pravidelně, ale více náhodně, což je přirozenější. Všechny tyto nepravidelnosti byly zadávány ručně pomocí dodatečných pravidel. Tři iterace pomohly k tomu, aby řasa vypadala delší, ale jak již bylo zmíněno v popisu funkcí a možností tohoto generátoru, tři iterace jsou v tomto případě minimální možný počet iterací.

¹⁰ Canvas představuje kreslící plátno.

¹¹ Scrollování je možnost použití posuvníku v okně.

Pravidla a nastavení rostliny:

- Axiom: F
- Délka čáry: 5
- Iterace: 3
- Počáteční úhel: -90°
- Úhel větvení: 25°

$$G \rightarrow FF[+FF[+FFF[-FFF]FF]FF]$$

$$A \rightarrow FF[-FFF]FFF[-FF[+FFF[-FFF]FF]FFF]$$

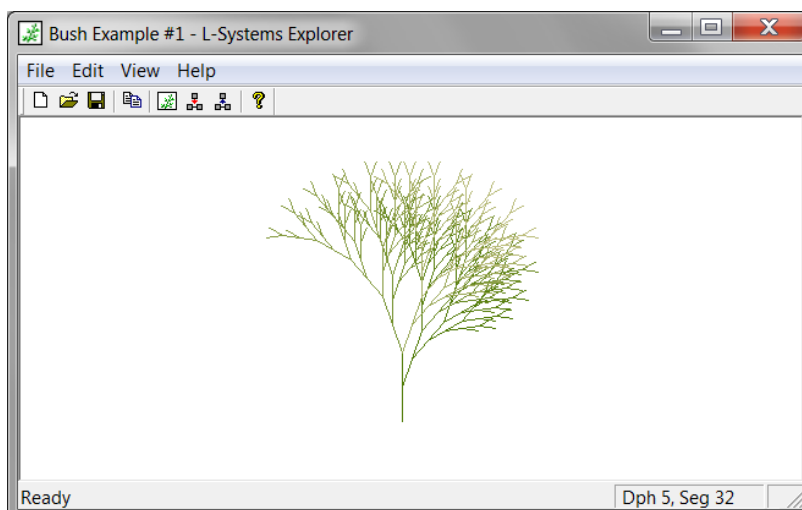
$$F \rightarrow G[-G[-GG]]G[+GG[-G[-A[+AA]G]A]F[-G]A]FF$$



Obr. č. 55 Vygenerovaná rostlina pomocí aplikace Lindenmayer System Generator.

4.4.2 L-Systems Explorer

Druhým předváděným generátorem je volně dostupný software, který nevyžaduje instalaci a opět generuje pouze deterministické L-systémy. Aplikaci lze získat z [Matthews, 2002]. Má možnost rostlinu obarvit dvěma barvami, což je přidáný efekt oproti předešlému online generátoru a L-systémová rostlina se tak zase o něco víc může přiblížit realitě.



Obr. č. 56 Náhled aplikace L-Systems Explorer.

Aplikace ukládá a načítá svůj vlastní formát souborů s příponou „lse“, umožňuje tak ukládat a znovu načítat vytvořená pravidla. Poskytuje několik ukázkových příkladů, aby se uživatel mohl podívat, jak vypadají prepisovací pravidla některých základních L-systémů a jak na ně program reaguje.

L-Systems Explorer umožňuje vytvářet 26 různých pravidel a obsahuje jednu velkou změnu oproti ostatním generátorům: pravidlo vykresluje rekurzivně [Matthews, 2002]. Jestliže písmeno „F“ představuje rovnou čáru a zároveň je to axiom, což znamená, že se do něj vepisuje posloupnost pravidel, tak se při každém napsaném „F“ vykoná celé pravidlo F. Tato funkce byla v některých případech nežádoucí, neboť cílem bylo kontrolovaně vytvořit jistý tvar rostliny, bez rekurzí. V takových situacích byl místo písmene „F“ použit znak, který generátor nabízí pro nakreslení jednoduché čáry bez rekurzí, tedy „|“.

V záložce „Edit“ je hned na prvním místě možnost otevření okna pro zadání L-systému a jeho nastavení. Jakmile je toto okno otevřeno, je původní okno neaktivní, to například znamená, že nelze levým tlačítkem posouvat rostlinu po plátně. V tomto okně s nastavením L-systému se zadává název systému, písmeno axiomu, úhel při otočení, počáteční úhel začátku generování, segment (celková velikost rostliny, čím větší číslo, tím větší rostlina), step (počet iterací, zadávají se hodnoty od 0-1) a nakonec samotná pravidla. Uživatelsky nepříjemné je zavření okna po každé, když se odsouhlasí nastavení tlačítkem „OK“, aby se nastavená změna vykreslila. Vyžaduje to neustálé otevírání okna s nastavením.

Pro tento software byly vygenerovány dvě ukázkové rostliny – keř (Obr. č. 57) a luční rostlina (Obr. č. 58). Keř byl generován s funkcí rekurzivního pravidla F. U luční rostliny již tato funkce využita nebyla a byl používán znak „|“.

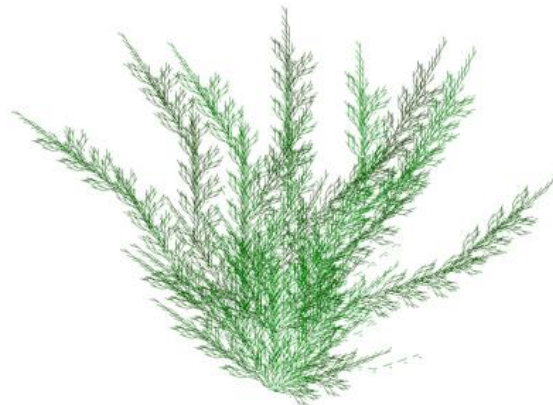
Pravidla a nastavení rostliny A:

- Axiom: F
- Angle: 25
- Initial angle: 270
- Segment: 6
- Step: 1

$$A \rightarrow FF[-FFF]FFF[-FF[+FFF[-FFF]FF]FFF]$$

$$G \rightarrow FF[+FF[+FFF[-FFF]FF]FF]$$

$$F \rightarrow FG[-G[-GG]]G[+GG[-G[-A[+AA]G]A]F[-G]A]FF$$



Obr. č. 57 Výsledný keř (A).

Pravidla a nastavení rostliny B:

- Axiom: F
- Angle: 8
- Initial Angle: 278
- Segment: 200
- Step: 0.5

$$B \rightarrow +[3+F]$$

$$F \rightarrow |[3+F][3-F]-|[4+F][6-F]+|[3-F][4-F]B-|F$$



Obr. č. 58 Výsledná rostlina (B).

U rostliny B je použit zkrácený zápis počtu znaků „+“ a „-“, který tento software podporuje. Místo například „+++F“ je zápis zkrácen na tvar „3+F“.

4.4.3 WWW L-System Explorer

Tento generátor je volně dostupná webová aplikace, která obsahuje i českou lokalizaci. Aplikace je k dispozici na [Kudrle, 2006a]. Výsledné L-systémy jsou deterministické, nicméně software má možnost generovat náhodně dlouhé kroky pomocí speciálního znaku „R“, tudíž vygenerovaná rostlina může vypadat při každém generování trochu jinak. Nejedná se ale o pravý stochastický L-systém, dojem stochastičnosti je pouze simulován. Níže je tato funkce demonstrována na testovacím modelu.

Ovládacích prvků má tento generátor mnoho a jsou v této kapitole diskutovány pouze ty nejdůležitější. Aplikace má dobře zpracovanou nápovědu ke všem prvkům, dostupnou na [Kudrle, 2006b].

Generátor umožňuje nastavení velikosti plátna (šířku a výšku v pixelech), což je novinka oproti ostatním generátorům. Standardní nastavení velikosti kreslicího plátna v tomto generátoru je 400×400 px.

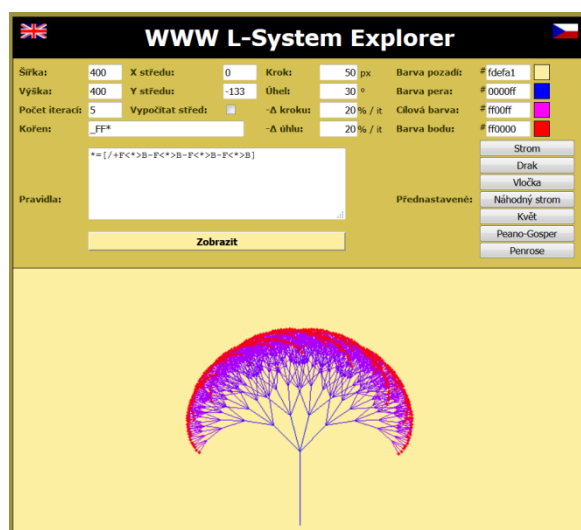
Další hodnoty, které lze nastavit jsou: Počet iterací, hodnota X a Y v plátně (X středu, Y středu), délka čáry (krok) v pixelech, úhel při otočení želvy (úhel), míra zkrácení délky čáry při vnořené iteraci <> v procentech (Δ kroku), míra zmen-

šení úhlu při vnořené iteraci $\langle \rangle$ v procentech (Δ úhlu), barva pozadí v šestnáctkovém tvaru, barva pera, cílová barva a barva bodu. K dispozici jsou i některé přednastavené L-systémy na ukázkou. Na Obr. č. 59 je ukázkou prvního přednastaveného L-systému, tedy „Strom“.

Hlavní částí jsou opět prepisovací pravidla, v tomto případě je zde na psaní pravidel pouze jeden box, ale pravidel lze zadávat víc. Pravidla se oddělují pomocí středníku „;“. Nejprve je opět nutno zvolit si kořen, který má v tomto generátoru tvar, který začíná vždy znakem podtržítka „_“, které říká „polož želvu na plátno“. Následující znaky jsou popsány z důvodu jejich použití na ukázkou L-systému.

Symbol	Význam symbolu
F	krok želvou dopředu
R	krok želvou dopředu o náhodné délce kroku
<	zanoření o jednu iteraci
>	vynoření o jednu iteraci
*	velká tečka
_	položít želvu

Tabulka 3 Používané symboly aplikace WWW L-System Explorer.



Obr. č. 59 Náhled aplikace WWW L-System Explorer.

Testovací subjekt pro tento generátor bylo vygenerování modelu keře šeríku (Obr. č. 60). Jak již bylo výše zmíněno, tento model je vytvořen za pomoci znaku

„R“, který generuje náhodně dlouhé kroky. Výsledný model tedy působí stochasticky.

Pravidla a nastavení rostliny:

- Kořen: _*
- Počet iterací: 7
- Krok: 30 px
- Úhel: 30°
- Δ kroku: 30 %
- Δ úhlu: 10 %

$*=[+RR<*>][RRRR<*>][-RR<*>]$



Obr. č. 60 Výsledné modely keře šeříku.

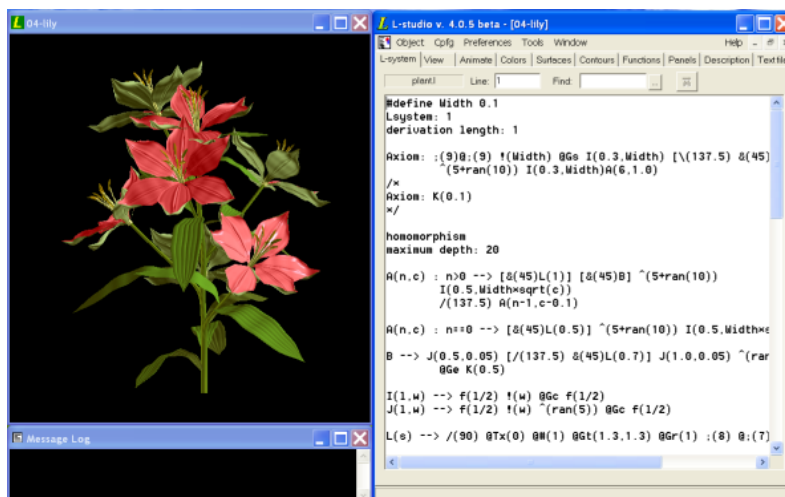
Subjektivní dojem z tohoto softwaru je velmi, přestože se jedná pouze o webovou aplikaci, je schopná generovat dobře vypadající modely rostlin a dalších fraktálních struktur.

4.4.4 L-Studio

L-Studio je volně dostupný software z [Algorithmic Botany], který výborně generuje L-systémy. V této práci byla použita verze ATE 4.2.13, která obsahuje všechny potřebné knihovny pro práci se softwarem. L-Studio nevyžaduje instalaci.

Nově, oproti předešlým představovaným softwarům, umí generovat i stochastické L-systémy. Po popisu programu bude k otestování vygenerován právě stochastický L-systém stromu.

Na první pohled může L-studio vypadat složitě a nepřehledně. K vygenerování základního 2D L-systému ale uživateli stačí používat pouze tři záložky, a to: „L-system“, „View“ a „Colors“.



Obr. č. 61 Náhled aplikace L-Studio.

Zdroj: [Algorithmic Botany]

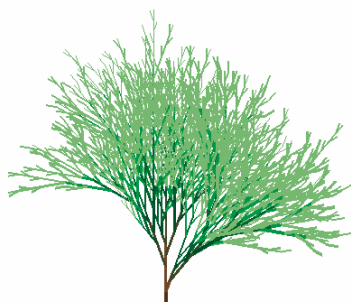
V záložce „L-System“ se zadává např. název L-systému, počet iterací, semínko, axiom a především přepisovací pravidla, kterými uživatel určuje podobu výsledného L-systému.

Následuje záložka „View“, do které se již nezapisují přepisovací pravidla, ale informace, jak daný L-systém bude zobrazen, informace o velikosti, použitých barvách, šířce čar a mnoho dalších.

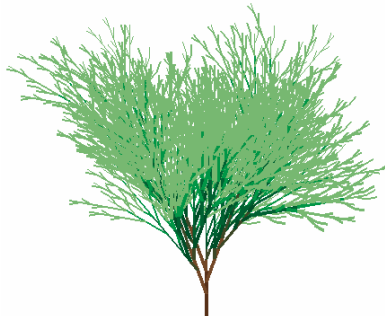
Čtvrtá záložka „Colors“ definuje barvy použitelné na obarvení určitých iterací L-systému. Nabízí 42 různých barev, které lze měnit na libovolné odstíny RGB. Jestliže byl v záložce „View“ parametr „color increment“ nastaven např. na hodnotu 1, tak při další iteraci bude pro L-systém vybrána následující barva. Nastavením hodnoty na 2 by byla použita vždy ob jedna barva atp.

Pro vygenerování sepsaného kódu je potřeba v hlavním menu otevřít záložku „Cpfg“ a aktivovat funkci „Go“. Jestliže je L-systém již jednou vygenerovaný, není nutné ho při každém dalším vyzkoušení zavřít funkcí „Kill“, stačí na konci nabídky funkcí v záložce „Cpfg“ kliknout na „Rerun“. Při testování bylo zjištěno, že ne všechny změny byly propagovány do vizualizace, takže pro jistotu je vhodnější okno s vizualizací zavřít a nechat ho znovu vygenerovat, aby byla výsledná vizualizace vždy aktuální. Při zaškrtnutí „Continuous modeling“ se některé změny provádí okamžitě.

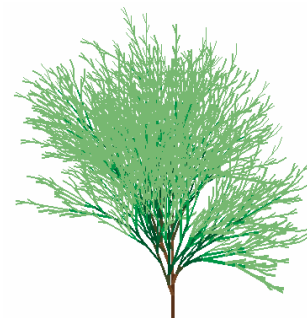
Pro otestování funkčnosti L-Studia byl vygenerován stochastický L-systém, připomínající strom nebo větší keř.



Obr. č. 62 Strom verze A.



Obr. č. 63 Strom verze B.



Obr. č. 64 Strom verze C.

Následující kód definuje předvedený stochastický L-systém na Obr. č. 62, Obr. č. 63 a Obr. č. 64. Všechny použité příkazy jsou vysvětleny v tabulce č. 4 pod zobrazeným kódem.

```
#define N 2.2

Lsystem: 1
derivation length: 7

seed: 2454

Axiom: F(1)
F(x) --> X(x*N) [+F(1)] F[-F(1)] F[+F(1)] !F(1) : 0.4
F(x) --> X(x) [-F(1)+F(1)] !F(1) : 0.6

homomorphism
X(x) --> ;F(x)

endlsystem
```

Kód	Význam
#define N 2.2	Definuje konstantu na hodnotu 2.2.
Lsystem: 1	Název L-systému.
derivation length: 4	Nastavuje počet iterací.
seed: 2454	Semínko náhodnosti pro náhodný generátor stochastických L-systémů.
Axiom: F(1)	Deklaruje Axiom na F(1).
F(x) --> X(x*N)[+F(1)...	Přepisovací pravidlo. V tomto L-systému jsou dvě stejná pravidla s levou stranou a jsou vážena pravdě-

	podobností na konci každého pravidla uvedenou za „:“.
homomorphism	Definuje interpretace symbolů. Např. symbol X se vykreslí jako F).
endsystem	Ukončovací příkaz.

Tabulka 4 Vysvětlení příkazů používaných k definování ukázkového L-systému.

Následující kód se nachází v záložce „View“. Tabulka č. 5 všechny použité příkazy vysvětluje.

```
scale factor: 0.7
angle increment: 20
initial color: 0
color increment: 1
initial line width: 4 pixels
line width increment: 0.8
```

Kód	Význam
scale factor: 0.7	Měřítko velikosti zobrazovaného L-systému v okně vizualizace.
angle increment: 20	Nastavuje úhel při větvení.
initial color: 0	Nastavuje počáteční barvu na hodnotu 0. Jsou používány hodnoty (jejich id) ze záložky Colors.
color increment: 1	Definuje o kolik barev se systém posune při iteraci.
initial line width: 4	Počáteční šířka vykreslované čáry. Zadáva se v pixelech.
line width increment: 0.8	O tuto hodnotu se zvětšuje/zmenšuje šířka používané čáry. Znak „#“ inkrementuje, znak „!“ dekrementuje.

Tabulka 5 Vysvětlení příkazů používaných k definování zobrazení ukázkového L-systému.

5 Shrnutí výsledků hodnocení

První předváděné aplikace byly založené na Perlinově šumové funkci. Nejpokročilejším generátorem je Filter Forge, plug-in pro Adobe Photoshop, jehož hlavní nevýhodou je vysoká cena, která se pohybuje od \$149 - \$828 (po přepočtu přibližně 3 500 - 20 000,- Kč) v závislosti na typu licence. Generuje vizuálně nejkvalitnější textury ze všech testovaných řešení, obsahuje nejvíce možností, parametrů a pokrývá největší škálu generovaných přírodních jevů. Pokud je cílem vygenerovat jednoduchou texturu s Perlinovým šumem, postačí Perlin Noise Generator, nebo TextureGenerator-Online. Jestliže je ale potřeba vygenerovat texturu na vyšší úrovni kvality a s větší možností úprav a zasahování do výsledné podoby, je nejlepší volbou plug-in Filter Forge pro Adobe Photoshop.

Fraktály byly předvedeny v aplikacích Mystica a Ultra Fractal. Ani jedna aplikace není volně dostupná, a tak doporučení pro rozhodování o použití softwaru na tomto parametru nezávisí. Fraktály lze jednoznačně lépe generovat aplikací Ultra Fractal. V té je možné zadat vlastní algoritmus pro podobu fraktálu, aplikace Mystica naproti tomu pouze generuje náhodný fraktál, kde uživatel ani neví, o jaký algoritmus se jedná. Ultra Fractal je tedy určen zkušenějším uživatelům, matematikům a těm, kteří chtějí mít generování fraktálu plně pod kontrolou a ovlivňovat ho od naprostého začátku. Mystica se naopak hodí pro uživatele, kteří nemají teoretické znalosti fraktálů a pro ty, kteří nechtějí zadávat fraktál ručně. Pro tyto uživatele může být aplikace Mystica užitečná.

Buněčné generování bylo vyzkoušeno v aplikacích Worley Texture Generator a v plug-inu Filter Forge pro Adobe Photoshop, který byl použit i pro Perlinovu šumovou funkci. I pro tuto metodu byl nejlepší plug-in Filter Forge, jelikož jeho výstupy, v podobě generovaných textur, vypadaly kvalitněji a s nastavením parametrů pro generování se dalo manipulovat ve větší míře než u aplikace Worley Texture Generator. Worley Texture Generator měl nedostatky při zadávání barev a hodnoty „0“, lze ho tedy doporučit jen pro demonstrační účely, nebo pro uživatele, kteří nemají k dispozici Filter Forge a Adobe Photoshop.

Zkoumanými aplikacemi na generování L-systémů byly Lindenmayer System Generator, L-Systems Explorer, WWW L-System Explorer a L-Studio. L-

systemy mají vizuálně atraktivnější výstupy, pokud umožňují stochastická pravidla, případně alespoň barevné alternace. Stochastické L-systemy z předvedených generátorů umí generovat pouze jeden software, L-Studio. Na začínající uživatele tento software působí složitě, ale snadnému použití pomáhá velmi dobře zpracovaný návod. L-Studio je volně dostupné, což je vzhledem k dlouhodobému vývoji, zaměření a širokým možnostem uplatnění překvapivé. Nejméně lze doporučit Lindenmayer System Generator, který umí generovat pouze černobílé deterministické L-systemy. Postačí ale na seznámení s L-systemy či demonstraci fungování této metody. Po L-Studio byl analyzován WWW L-System Explorer, který obsahuje dokonce i českou lokalizaci. Generátor sice nebyl schopen generovat pravé stochastické L-systemy, které umožňují produkovat zajímavější výsledky, ale umí některé vlastnosti stochastických L-systémů napodobit svým unikátním symbolem „R“, který generuje náhodně dlouhé kroky, čímž do jisté míry odstraňuje nechtěnou pravidelnost. L-System Explorer v sobě má zabudovanou funkci, že se symbol F bude vykreslovat rekurzivně, čehož se dá využít, avšak u mnoha L-systémů je tato vlastnost spíše na obtíž. V případě použití symbolu F jsou výsledky výrazně odlišné od nerekurzivní verze a může být složité tuto chybu odhalit. Tento rekurzivní symbol lze nahradit symbolem „|“, který již kreslí klasickou čáru, bez rekurze.

6 Závěr

V rámci práce byly představeny často využívané metody procedurálního generování textur. Zmíněnými metodami byly: Perlinova šumová funkce, fraktály, buněčné texturování a L-systémy. Nastudované teoretické informace byly otestovány v existujících aplikacích, čímž byly otestovány možnosti praktického použití analyzovaných metod pro procedurální generování textur.

Tato práce byla zaměřena na popis metod pro procedurální generování textur a následné vyzkoušení těchto metod v praxi. Aplikace byly hodnoceny jak objektivně podle funkčnosti, možností, složitosti, dostupnosti, jejich limitů, tak subjektivně podle uživatelské přívětivosti. Výsledný přehled metod a jejich aplikací může soužit pro lepší pochopení těchto metod a orientaci pro rozhodování, které metody jsou vhodné pro dané typy textur. Český jazyk této práce by zároveň měl odstraňovat bariéru pro uživatele, kteří neumí cizí jazyk, ale přesto by chtěli porozumět probíraným metodám a aplikacím.

Popisované metody a většina aplikací byly soustředěné na dvourozměrný prostor (2D). Procedurální textury lze však rozšířit například do trojrozměrného prostoru (3D) či procedurálně generovat animované textury, což bude předmětem dalšího výzkumu.

7 Seznam použité literatury

ALGORITHMIC BOTANY. *Algorithmic Botany: Software* [online]. [Cit. 25.3.2016a]. Dostupné z WWW: <http://algorithmicbotany.org/virtual_laboratory/>.

ALGORITHMIC BOTANY. *L-studio User's Guide: What is L-studio?* [online]. [Cit. 25.3.2016b]. Dostupné z WWW: <<http://algorithmicbotany.org/lstudio/whatis.html>>.

ALLENE, C., PONS, J.-P. & KERIVEN, R. Seamless image-based texture atlases using multi-band blending. In *19th International Conference on Pattern Recognition, 2008. ICPR 2008*. 2008, s. 1–4.

ALLEN, P. *Modeling Plants with Lindenmayer Systems - Allen Pike* [online]. 2007. [Cit. 6.3.2016]. Dostupné z WWW: <<http://www.allenpike.com/modeling-plants-with-l-systems/>>.

ALL GRANITE AND MARBLE CORP. *Arabescato Marble Veiny Vanity Material, Black, White, Grey*. [online]. 2015. [Cit. 20.3.2016]. Dostupné z WWW: <<http://marble.com/material/marble/202/arabescato#>>.

ECG WIKI. *2D Perlin Noise Texture* [online]. 2011. [Cit. 5.3.2016]. Dostupné z WWW: <https://lva.cg.tuwien.ac.at/ecg/wiki/doku.php?id=students:perlin_noise>.

AURENHAMMER, F. & KLEIN, R. Voronoi Diagrams. In *Handbook of Computational Geometry*. Amsterdam ; New York : North-Holland, 2000. ISBN 978-0-444-82537-7.

BEUTEL, A. *Interactive Voronoi Diagram Generator with WebGL - Alex Beutel* [online]. 2015. [Cit. 27.3.2016]. Dostupné z WWW: <<http://alexbeutel.com/webgl/voronoi.html>>.

BOURKE, P. *L-System bush* [online]. [Cit. 10.2.2015]. Dostupné z WWW: <http://paulbourke.net/fractals/lsys_bush_b/>.

BUTRAM, T. *Tutorial 4: Modifying noise modules — pynoise 2.1.0 documentation* [online]. 2015. [Cit. 3.3.2016]. Dostupné z WWW: <<http://pynoise.readthedocs.org/en/latest/tutorial4.html>>.

COHABITAIRE. *Natural Fractals / Cohabitaire - Lifestyle inspired by nature* [online]. [Cit. 29.3.2016]. Dostupné z WWW: <<http://www.cohabitaire.com/natural-fractals/>>.

CONVERSANO, E. & TEDESCHINI-LALLI, L. Sierpinski Triangles in Stone on Medieval Floors in Rome. *Journal of Applied Mathematics*, 2011, vol. 4, no. 4.

DAWNTEC. *Mystica 2D and 3D fractal generator with editor for art and texture*. [online]. [Cit. 24.3.2016]. Dostupné z WWW: <<http://www.dawntec.com/mystica/>>.

DŘEVO SPIRIT OF ANGMAR & BITHYNIE. *Dřevo - BUK* [online]. [Cit. 29.3.2016]. Dostupné z WWW: <<http://prace-se-drevem.spibi.cz/Drevo-Druhy-Buk.html>>.

FILTER FORGE, INC. *Filter Forge - Photoshop Plugin to Create Your Own Filters* [online]. 2016. [Cit. 12.4.2016]. Dostupné z WWW: <<https://www.filterforge.com/>>.

FLIGHTGEAR WIKI. *Procedural Texturing - FlightGear wiki* [online]. 2014. [Cit. 10.2.2015]. Dostupné z WWW: <http://wiki.flightgear.org/Procedural_Texturing>.

GALLERYHIP. *Clay Texture Seamless* [online]. [Cit. 10.2.2015]. Dostupné z WWW: <<http://galleryhip.com/clay-texture-seamless.html>>.

GUSTAVSON, S. *Simplex noise demystified* [online]. 2005. [Cit. 8.2.2015]. Dostupné z WWW: <<http://webstaff.itn.liu.se/~stegu/simplexnoise/simplexnoise.pdf>>.

HARTL, A. *RunicSoft: NoiseGen Perlin Noise Generator* [online]. 2011. [Cit. 22.3.2016]. Dostupné z WWW: <<http://www.runicsoft.com/noisegen.php>>.

HUGO, E. *Perlin Noise* [online]. 2000. [Cit. 4.3.2016]. Dostupné z WWW: <http://freespace.virgin.net/hugo.elias/models/m_perlin.htm>.

KUDRLE, Z. *WWW L-System Explorer* [online]. 2006a. [Cit. 23.3.2016]. Dostupné z WWW: <<http://zdeck.borg.cz/wlse/l-system.php>>.

KUDRLE, Z. *WWW L-System Explorer Help* [online]. 2006b. [Cit. 23.3.2016]. Dostupné z WWW: <<http://zdeck.borg.cz/wlse/help.php?lang=cz#iter>>.

MANDELBROT, B.B. *The Fractal Geometry of Nature /Revised and Enlarged Edition/*. New York, W.H. Freeman and Co., 1983, 495 P., 1983, vol. -1.

MATTHEWS, J. *generation5 - L-Systems Explorer* [online]. 2002. [Cit. 23.3.2016]. Dostupné z WWW: <<http://www.generation5.org/content/2002/lse.asp>>.

MCCOMBS, S. *Intro to Procedural Textures* [online]. [Cit. 10.2.2015]. Dostupné z WWW: <<http://www.upvector.com/?section=Tutorials&subsection=Intro%20to%20Procedural%20Textures>>.

MCNALLY, J. *Earth's Most Stunning Natural Fractal Patterns* [online]. 2010, WIRED. [Cit. 29.3.2016]. Dostupné z WWW: <<http://www.wired.com/2010/09/fractal-patterns-in-nature/>>.

MENGER, K. *Dimensionstheorie*. Springer Fachmedien Wiesbaden, 1928. ISBN 978-3-663-15484-6.

NOLAND, C. *Lindenmayer System Generator* [online]. [Cit. 23.3.2016]. Dostupné z WWW: <<http://nolandc.com/sandbox/fractals/?x=550&y=714&l=8&d=3&sa=-90&a=22&s=F&r=,F,F>>.

NOODLE NUKE. *The Earth's 10 Most Insane Natural Fractal Patterns* [online]. 2015, NoodleNuke. [Cit. 29.3.2016]. Dostupné z WWW: <<http://noodlenuke.com/the-earths-most-insane-natural-fractal-patterns/9655>>.

NOWELL, W.L. *Fractals - Where Math and Art Collide* [online]. 2013a. [Cit. 12.4.2016]. Dostupné z WWW: <<http://www.cmath.info/html/fractalArt.html>>.

NOWELL, W.L. *Resources for Undergraduate Mathematics Students* [online]. 2013b. [Cit. 29.3.2016]. Dostupné z WWW: <<http://www.cmath.info/html/kochPerimeter.html>>.

PERLIN, K. An image synthesizer. *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985, vol. 19, no. 3, s. 287–296. ISSN 0097-8930.

PERLIN, K. Improving Noise. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '02. New York, NY, USA : ACM, 2002, s. 681–682. ISBN 1-58113-521-1.

PETRY, C. *TextureGenerator-Online* [online]. [Cit. 21.3.2016]. Dostupné z WWW: <<http://cpetry.github.io/TextureGenerator-Online/>>.

PROAKIS, J.G. & MANOLAKIS, D.G. *Digital Signal Processing: Principles, Algorithms & Applications* Third Edition. USA : Prentice-Hall, Inc., 1996. ISBN 0-13-394338-9.

PRUSINKIEWICZ, P., L., A. *The Algorithmic Beauty of Plants*. Springer Science & Business Media, 2004.

REICHL, J. & VŠETIČKA, M. *Encyklopedie fyziky* [online]. 2006. [Cit. 7.2.2015]. Dostupné z WWW: <<http://fyzika.jreichl.com/main.article/view/15-pohyb-hmotneho-bodu-po-kruznicu>>.

ROCKPANEL GROUP. *Rockpanel Woods boards can scarcely be differentiated from real wood* [online]. [Cit. 20.3.2016]. Dostupné z WWW: <<http://www.rockpanel.co.uk/products/rockpanel+woods>>.

SLIJKERMAN, F. *Ultra Fractal: Download* [online]. 2010. [Cit. 24.3.2016]. Dostupné z WWW: <<http://www.ultrafractal.com/download/>>.

SOLTANIFAR, M. On A sequence of cantor Fractals. *Rose Hulman Undergraduate Mathematics Journal*, 2006, vol. 7, no. 1.

SOMETHINGHITME. *Worley Texture Generator* [online]. 2010. [Cit. 27.3.2016]. Dostupné z WWW: <<http://somethinghitme.com/projects/cell/>>.

TAN, L. Similarity between the Mandelbrot set and Julia Sets. *Communications in Mathematical Physics*, 1990, no. 134.

TANNER, P.P., JOLICOEUR, P., COWAN, W.B., BOOTH, K. & FISHMAN, F.D. Antialiasing: A Technique for Smoothing Jagged Lines on a Computer Graphics

Image—an Implementation on the Amiga. *Behavior Research Methods, Instruments, & Computers*, 1989, vol. 21, no. 1, s. 59–66. ISSN 0743-3808, 1532-5970.

TIŠNOVSKÝ. *Procedurální textury a role turbulence* [online]. 2008, Root.cz. [Cit. 7.2.2015]. Dostupné z WWW: <<http://www.root.cz/clanky/proceduralni-textury-a-role-turbulence/>>.

TIŠNOVSKÝ, P. *L-systémy: přírodní objekty i umělé artefakty* [online]. 2006a, Root.cz. [Cit. 29.3.2016]. Dostupné z WWW: <<http://www.root.cz/clanky/l-systemy-prirodni-objekty-i-umele-artefakty/>>.

TIŠNOVSKÝ, P. *Závorkové a stochastické L-systémy - Root.cz* [online]. 2006b. [Cit. 11.3.2016]. Dostupné z WWW: <<http://www.root.cz/clanky/zavorkove-a-stochasticke-l-systemy/>>.

WORLEY, S. A Cellular Texture Basis Function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '96. New York, NY, USA : ACM, 1996, s. 291–294. ISBN 0-89791-746-4.

ŽÁRA, J., BENEŠ, B., SOCHOR, J. & FELKEL, P. *Moderní počítačová grafika 2. Přepřacované a rozšířené vydání*. Brno : Computer Press, a. s., 2010. ISBN 80-251-0454-0.

8 Seznam obrázků

OBR. Č. 1 PŘÍKLAD BEZEŠVÉ TEXTURY, ROZPRASKANÁ ZTVRDLÁ HLÍNA.....	2
OBR. Č. 2 ČTYŘI STEJNÉ BEZEŠVÉ TEXTURY VEDLE SEBE VYTVÁŘEJÍ VĚTŠÍ PLOCHU BEZ VIDITELNÉHO NAPOJENÍ TEXTUR.....	2
OBR. Č. 3 KLASICKÁ TEXTURA, PROBLÉM S PŘECHODY A OPAKOVÁNÍM.....	3
OBR. Č. 4 POUŽITÍ PROCEDURÁLNÍ TEXTURY, ODSTRAŇUJE PROBLÉM S PŘECHODY A OPAKOVÁNÍM.	3
OBR. Č. 5 KLASICKÁ TEXTURA, PROBLÉM S ROZLIŠENÍM.....	3
OBR. Č. 6 POUŽITÍ PROCEDURÁLNÍ TEXTURY, ODSTRAŇUJE PROBLÉM S ROZLIŠENÍM.....	3
OBR. Č. 7 PRAVIDELNÁ MŘÍŽKA S NÁHODNĚ SMĚROVANÝMI 2D VEKTORY PRO KAŽDÝ CELOČÍSELNÝ BOD V MŘÍŽCE.....	5
OBR. Č. 8 HLADKÁ INTERPOLACE MEZI PRAVIDELNĚ USPOŘÁDANÝMI BODY NA OSE (1D).....	6
OBR. Č. 9 ŠUMOVÉ FUNKCE PŘED SOUČTEM S RŮZNÝMI HODNOTAMI AMPLITUD A FREKVENCÍ.....	6
OBR. Č. 10 ŠUMOVÉ FUNKCE PO SOUČTU.....	7
OBR. Č. 11 DVOJROZMĚRNÉ RŮZNÉ ŠUMOVÉ FUNKCE PŘED SOUČTEM.....	7
OBR. Č. 12 DVOJROZMĚRNÁ ŠUMOVÁ FUNKCE PO SOUČTU.....	7
OBR. Č. 13 VLIV PERSISTENCE NA VÝSTUP PERLINOVY ŠUMOVÉ FUNKCE.....	8
OBR. Č. 14 $p = 0.25$	8
OBR. Č. 15 $p = 0.5$	8
OBR. Č. 16 $p = 0.75$	8
OBR. Č. 17 VLIVY TURBULENCE NA PŮVODNÍ PRAVIDELNÉ PRUHY, VZNIKÁ TEXTURA MRAMORU. ..	9
OBR. Č. 18 UKÁZKA TVARU PRO 2D PROSTOR.....	10
OBR. Č. 19 BODY VE 2D PROSTORU – VÝSLEDEK SE ZÍSKÁVÁ SEČTENÍM POUZE ČERVENĚ ZAKROUŽKOVANÝCH BODŮ.....	11
OBR. Č. 20 POROVNÁNÍ PŮVODNÍ PERLINOVY ŠUMOVÉ FUNKCE (VLEVO) SE SIMPLEXOVÝM ŠUMEM (VPRAVO) PRO 2D.....	11
OBR. Č. 21 STARÁ (ČERNÁ) A NOVÁ (ČERVENÁ) INTERPOLAČNÍ KŘIVKA.....	12
OBR. Č. 22 POUŽITÍ STARÉ (VLEVO) A NOVÉ (VPRAVO) INTERPOLACE.....	12
OBR. Č. 23 ČTYŘI ITERACE KOCHOVY VLOČKY.....	14
OBR. Č. 24 BROKOLICE ROMANESCO.....	14
OBR. Č. 25 ULITA HLEMÝŽDĚ.....	14

OBR. Č. 26 SLUNEČNICE.	14
OBR. Č. 27 PÁV A JEHO PEŘÍ.....	14
OBR. Č. 28 VORONOIŮV DIAGRAM.	15
OBR. Č. 29 VIZUÁLNÍ PODOBA FUNKCÍ F1, F2 A JEJICH VARIANT.	16
OBR. Č. 30 KEŘ VYGENEROVANÝ POMOCÍ L-SYSTÉMŮ.	17
OBR. Č. 31 DETERMINISTICKÉ 3D ROSTLINY VYGENEROVANÉ POMOCÍ L-SYSTÉMŮ.	19
OBR. Č. 32 (A) ŽELVÍ INTERPRETACE SYMBOLŮ F , $+$, $-$. (B) INTERPRETACE POHYBU ŽELVY PODLE SYMBOLŮ. ÚHEL OTOČENÍ ŽELVY JE ROVEN 90° . NA POČÁTKU ŽELVA SMĚŘUJE NAHORU.	20
OBR. Č. 33 GRAFICKY ZNÁZORNĚNÝ ZÁVORKOVÝ L-SYSTÉM S PRAVIDLY.	21
OBR. Č. 34 NÁHLED APLIKACE PERLIN NOISE GENERATOR.....	23
OBR. Č. 35 NÁHLED APLIKACE TEXTUREGENERATOR-ONLINE.....	24
OBR. Č. 36 NÁHLED PLUG-INU FILTER FORGE.....	25
OBR. Č. 37 UKÁZKA PRVKŮ, ZE KTERÝCH SE SKLÁDÁ VÝSLEDNÁ TEXTURA.....	26
OBR. Č. 38 PŘEDLOHA BUKOVÉHO DŘEVA.....	26
OBR. Č. 39 VÝSLEDNÁ TEXTURA BUKOVÉHO DŘEVA VYTVOŘENÉHO PŘES FILTER FORGE.....	26
OBR. Č. 40 PŘEDLOHA MRAMORU.	27
OBR. Č. 41 VÝSLEDNÁ TEXTURA MRAMORU VYTVOŘENÁ PŘES FILTER FORGE.	27
OBR. Č. 42 NÁHLED APLIKACE MYSTICA.	28
OBR. Č. 43 VERZE A TEXTURY S EFEKTY.	29
OBR. Č. 44 VERZE B TEXTURY S EFEKTY.	29
OBR. Č. 45 VERZE C TEXTURY S EFEKTY.....	29
OBR. Č. 46 PŮVODNĚ VYGENEROVANÁ TEXTURA BEZ EFEKTŮ.	29
OBR. Č. 47 NÁHLED APLIKACE ULTRA FRACTAL S MANDELBROTOVOU MNOŽINOU.	30
OBR. Č. 48 BARNSLEY (ZÁKLADEM JE JULIOVA MNOŽINA).	31
OBR. Č. 49 PRVNÍ ZMĚNA PŮVODNÍHO PŘEDPISU.....	32
OBR. Č. 50 DRUHÁ ZMĚNA PŮVODNÍHO PŘEDPISU.	32
OBR. Č. 51 VÝSLEDNÝ FRAKTÁL PO ZMĚNÁCH.	32
OBR. Č. 52 NÁHLED APLIKACE WORLEY TEXTURE GENERATOR.....	33
OBR. Č. 53 VÝSLEDNÉ UKÁZKOVÉ TEXTURY A (VLEVO) A B (VPRAVO).	34
OBR. Č. 54 VÝSLEDNÉ TEXTURY KAMENE (VLEVO) A MIKROSKOPICKÝCH BAKTERIÍ (VPRAVO).....	35
OBR. Č. 55 VYGENEROVANÁ ROSTLINA POMOCÍ APLIKACE LINDENMAYER SYSTEM GENERATOR.	37
OBR. Č. 56 NÁHLED APLIKACE L-SYSTEMS EXPLORER.....	38

OBR. Č. 57 VÝSLEDNÝ KEŘ (A).....	39
OBR. Č. 58 VÝSLEDNÁ ROSTLINA (B).	40
OBR. Č. 59 NÁHLED APLIKACE WWW L-SYSTEM EXPLORER.....	41
OBR. Č. 60 VÝSLEDNÉ MODELY KEŘE ŠEŘÍKU.....	42
OBR. Č. 61 NÁHLED APLIKACE L-STUDIO.....	43
OBR. Č. 62 STROM VERZE A.	44
OBR. Č. 63 STROM VERZE B.	44
OBR. Č. 64 STROM VERZE C.	44

9 Seznam tabulek

TABULKA 1: ZÁKLADNÍ SYMBOLY ŽELVÍ GRAFIKY.....	19
TABULKA 2: NASTAVENÍ VÝSLEDNÝCH TEXTUR VYGENEROVANÝCH APLIKACÍ WORLEY TEXTURE GENERATOR.	34
TABULKA 3: POUŽÍVANÉ SYMBOLY APLIKACE WWW L-SYSTEM EXPLORER.....	41
TABULKA 4: VYSVĚTLENÍ PŘÍKAZŮ POUŽÍVANÝCH K DEFINOVÁNÍ UKÁZKOVÉHO L-SYSTÉMU.....	45
TABULKA 5: VYSVĚTLENÍ PŘÍKAZŮ POUŽÍVANÝCH K DEFINOVÁNÍ ZOBRAZENÍ UKÁZKOVÉHO L- SYSTÉMU.	45



FIM UHK

UNIVERZITA HRADEC KRÁLOVÉ

Fakulta informatiky a managementu

Rokitanského 62, 500 03 Hradec Králové, tel: 493 331 111, fax: 493 332 235

Zadání k závěrečné práci

Jméno a příjmení studenta:

Pavčina Kuřeová

Obor studia:

Aplikovaná informatika

Jméno a příjmení vedoucího práce:

Karel Petránek

Název práce:

Procedurální generování textur

Název práce v AJ:

Procedural Texture Generation

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Prozkoumat oblast procedurálního generování textur, popsat dosavadní metody, porovnat je a zhodnotit možnosti a rozsah jejich použití v 3D scénách.

Osnova práce:

- 1) Seznámení s problematikou
- 2) Průzkum existujících metod a jejich popis
- 3) Volba či implementace metod
- 4) Testování metod

Projednáno dne: 16. 10. 2014

Podpis studenta

Podpis vedoucího práce