



## **Středoškolská technika 2010**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

# **VYKRESLOVÁNÍ A NÁVRH LINDENMEYEROVÝCH SYSTÉMŮ**

**Michal Genserek**

**Střední průmyslová škola elektrotechniky a informatiky, Ostrava, příspěvková organizace  
Kratochvílova 7/1490, Ostrava - Moravská Ostrava**

# STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor 18 – Informatika

## Vykreslování a návrh Lindenmeyerových Systémů

Vypracoval:

**Michal Genserek**

**T4B**

Konzultant:

**Ing. Boleslav Martiník**

*Střední průmyslová škola  
elektrotechniky a informatiky  
Ostrava, 2010*

## **Anotace**

Program FracDraw je pokročilý unikátní editor a prohlížeč Lindenmayerových systémů, které jsou předpisem pro vykreslování fraktálních obrazců, nebo také pro modelování rostlin, stromů a jiných botanických fraktálních útvarů. Software se schopností vykreslování, popřípadě návrhu L-Systémů již existují různé druhy, avšak jakmile se začne člověk o problematiku více zajímat, narazí na poměrnou funkční omezenost nebo malou výkonost těchto programů což vede k nepraktičnosti použití těchto programů k výuce a pochopení vlastností a zákonitostí L-Systémů. Tyto důvody spolu se zájmem o fraktální grafiku mě vedly k vytvoření programu, který by byl svobodný a především plně uživatelsky přívětivý a snadný na použití a na druhou stranu by poskytoval pokročilá nastavení a funkce.

## **Prohlášení**

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, SW atd.) uvedenou v příloženém seznamu.

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským v platném znění.

V Ostravě dne .....podpis: .....

## Obsah

<b>1</b>	<b>Seznam obrázků.....</b>	<b>5</b>
<b>2</b>	<b>Úvod.....</b>	<b>6</b>
<b>3</b>	<b>Pojmy a vysvětlivky.....</b>	<b>7</b>
<b>4</b>	<b>Analýza.....</b>	<b>9</b>
<b>4.1</b>	<b>Úvod do problematiky fraktálů a L-Systemů.....</b>	<b>9</b>
<b>4.2</b>	<b>Soběpodobný fraktál.....</b>	<b>10</b>
<b>4.3</b>	<b>Praktické využití fraktálních obrazců.....</b>	<b>11</b>
<b>4.4</b>	<b>Princip funkce L-Systemů.....</b>	<b>11</b>
<b>4.5</b>	<b>Přepisovací pravidla a řídicí znaky.....</b>	<b>12</b>
<b>5</b>	<b>Implementace.....</b>	<b>13</b>
<b>5.1</b>	<b>Vývojová platforma a prostředí.....</b>	<b>13</b>
<b>5.2</b>	<b>Požadavky na GUI programu.....</b>	<b>14</b>
<b>5.3</b>	<b>Požadavek na grafický návrh předpisu L-Systemu.....</b>	<b>14</b>
<b>6</b>	<b>Základní GUI programu.....</b>	<b>15</b>
<b>6.1</b>	<b>Vykreslovací plocha.....</b>	<b>16</b>
<b>6.2</b>	<b>Panel s vlastnostmi fraktálu.....</b>	<b>17</b>
<b>6.3</b>	<b>Dialog „Správce nastavení“.....</b>	<b>18</b>
<b>6.4</b>	<b>Dialogy „Export“.....</b>	<b>19</b>
<b>6.5</b>	<b>Dialog „Editace“.....</b>	<b>20</b>
<b>6.6</b>	<b>Dialog „Editor palety“.....</b>	<b>22</b>
<b>6.7</b>	<b>Dialog „Editor pohybových pravidel“.....</b>	<b>23</b>
<b>6.8</b>	<b>Průvodce vytvořením nového L-Systemu.....</b>	<b>24</b>
<b>6.9</b>	<b>Statusbar.....</b>	<b>25</b>
<b>7</b>	<b>Závěr.....</b>	<b>27</b>
<b>7.1</b>	<b>Souhrn.....</b>	<b>27</b>
<b>7.2</b>	<b>Budoucnost projektu.....</b>	<b>27</b>
<b>8</b>	<b>Seznam použitých zdrojů.....</b>	<b>28</b>
<b>9</b>	<b>Použitý software.....</b>	<b>29</b>

## **1 Seznam obrázků**

<i>Obrázek 1. Princip měření Korsiky</i>	9
<i>Obrázek 2. Přírodní fraktální útvary</i>	10
<i>Obrázek 3. Soběpodobnost fraktálních obrazců</i>	10
<i>Obrázek 4. Konečný vzhled hlavního okna programu</i>	15
<i>Obrázek 5. Zoom panel</i>	16
<i>Obrázek 6. Panel vlastností</i>	17
<i>Obrázek 7. Správce nastavení</i>	18
<i>Obrázek 8. Export do rastru</i>	19
<i>Obrázek 9. Dialog Editace textového řetězce</i>	20
<i>Obrázek 10. Dialog Editace</i>	21
<i>Obrázek 11. Editor Palety</i>	22
<i>Obrázek 12. Dialog výběru barvy poskytovaný Java API</i>	22
<i>Obrázek 13. Editor znaků</i>	23
<i>Obrázek 14. Výchozí statusbar</i>	26
<i>Obrázek 15. Příklad uživatelsky upraveného statusbaru</i>	26

## 2 Úvod

Fraktální obrazce mě vždycky lákaly a přitahovaly, zejména svou na první pohled velikou složitostí a také svou doslova „nekonečnou“ krásou. V okamžiku, kdy se člověk rozhodne že by kromě obdivování se těmto obrazcům, chtěl také podobné vytvořit, začne se obvykle shánět po nějakém software který danou problematiku zvládne a umožní prozkoumat. V dnešní době sdílení informací na internetu najít takovýto program už není velký problém. Avšak najít pořádný program, jež zvládne jak základní tak pokročilé editace, nastavení a zobrazování a současně bude uživatelsky příjemný, rychlý, bez chyb a nejlépe zadarmo, se pro mne ukázalo jako neřešitelný problém. Rozhodl jsem se tedy program, co nejvíce odpovídající výše zmíněným vlastnostem sám napsat. Brzo se ukázalo že udělat program schopný zobrazovat různé druhy fraktálních obrazců je nad síly jednotlivce, ba i skupiny programátorů. Rozhodl jsem se tedy při vývoji programu zaměřit na problematiku L-Systemů na jejichž zobrazení, nemluvně o jejich editaci či úpravách, mnoho odborného software neexistuje. Většinou se jedná o různé „jednotýdenní“ záležitosti studentů programování kde si může, vzhledem k době vývoje, případně kvůli „odfláklosti“, uživatel nechat zdát o pokročilých funkcích a nastaveních. Můj program se snaží o vytvoření standardního interpretačního enginu pro vykreslování L-Systemů a zajistit tak zájemcům o fraktální obrazce a jejich tvorbu jednoduché uživatelské rozhraní spolu s množstvím pokročilých nastavení a možností.

Jelikož vykreslování L-Systemů pomocí takzvané želví grafiky je již dobře popsanou technikou, bude se tato práce zabývat spíše než metodami implementace analýzou a návrhem vývoje grafického uživatelského prostředí a taktéž inovačními myšlenkami pro vývoj L-Systemů v tomto programu.

## **3 Pojmy a vysvětlivky**

### **L-Systémy**

L-Systémy jsou rekurzivní přepisovací systémy. Více o nich v kapitole 4.

### **GUI**

Zkratka pro grafické uživatelské prostředí. Je to takové prostředí, které umožňuje ovládat počítačový program pomocí grafických uživatelských prvků.

### **OpenSource Software**

Opensource je počítačový software s otevřeným zdrojovým kódem. Otevřenost zde znamená jak technickou dostupnost kódu, tak legální dostupnost - licenci software, která umožňuje, při dodržení jistých podmínek, uživatelům zdrojový kód využívat, například prohlížet a upravovat.

### **Java**

Objektově orientovaný programovací jazyk vyvinutý firmou Sun v roce 1994. Hlavní výhodou tohoto jazyka je, že programy v něm napsané běží na libovolném operačním systému či platformě, pro kterou existuje Java VM, což činí z programů napsaných v jazyce Java jedny z nejvíce přenositelných.

### **Toolbar**

Je jedním z grafických uživatelských prvků, který poskytuje uživateli přístup k základním operacím programu. Jeho hlavní charakteristikou je, že můžeme tímto prvkem hýbat a přemístit jej tam kde zabírá v současné chvíli nejméně místa. Toolbary se typicky používají zejména v grafických a textových editorech, nebo třeba také v internetových prohlížečích.



## **Statusbar**

Je jedním z grafických uživatelských prvků, který bývá umístěn zpravidla v dolní části programu. Jeho úkolem je ukazovat uživateli informace o aktuálních operacích a stavu programu a jeho komponent.

## **Vektorová x Rastrová grafika**

Jsou 2 možné varianty uložení počítačové grafiky. Zatímco rastrová grafika uchovává informace jako mřížku bodů, kde má každý bod v této mřížce jinou barvu, vektorová grafika ukládá grafické informace vyjádřené pomocí základních geometrických objektů jako například bod, úsečka, kružnice a n-úhelník.

## **SVG**

SVG anebo také *Scalable Vector Graphics* je opensource formát pro popis vektorové grafiky za pomoci značkovacího jazyka XML.

## 4 Analýza

### 4.1 Úvod do problematiky fraktálů a L-Systemů

Na první pohled jsou fraktály jedny z nejsložitějších geometrických objektů, které současná matematika zkoumá avšak princip jejich struktury je z matematického případně geometrického hlediska poměrně jednoduchý. Poprvé použil termín fraktál francouzský matematik Benoit B. Mandelbrot v roce 1975 ačkoliv fraktální obrazce byly známy již dlouhou dobu předtím. Pokud chceme přesně definovat co je to fraktál zjistíme že máme smůlu. Vědci se stále nemohou shodnout na matematicky či vědecky přesné definici fraktálního obrazce. Laicky a značně zjednodušeně můžeme říci že fraktál je nekonečně členitý geometrický útvar. Další pololaická definice fraktálu zní že fraktál je obrazec s nekonečným obvodem. Možná si říkáte jak je toto možné, ale tato definice vychází z historického pokusu o změření obvodu ostrova Korsika. Jak přesně měření Korsiky probíhalo můžeme vidět na následujícím obrázku.



Obrázek 1. Princip měření Korsiky

Kartograf provádějící měření Korsiky zjistil, že pokaždé když zvýší přesnost měření (tedy počet úseček) vyjde mu větší obvod. Teoreticky vzato můžeme přesnost stále zvyšovat a dostat se tak až na úroveň subatomárních částic. Že je přírodní útvar fraktálem není nic zajímavého (alespoň ne moc :)), každý určitě již nějaký ten fraktál viděl. Alespoň každý kdo jí trochu zeleniny a nebo chodí do horských prostředí. Fraktály můžeme dělit na několik druhů. A právě pro popis výše jmenovaných zejména přírodních fraktálů používáme pojem Lindenmeyerovy Systémy (dále jen L-Systémy). Byly objeveny a poprvé

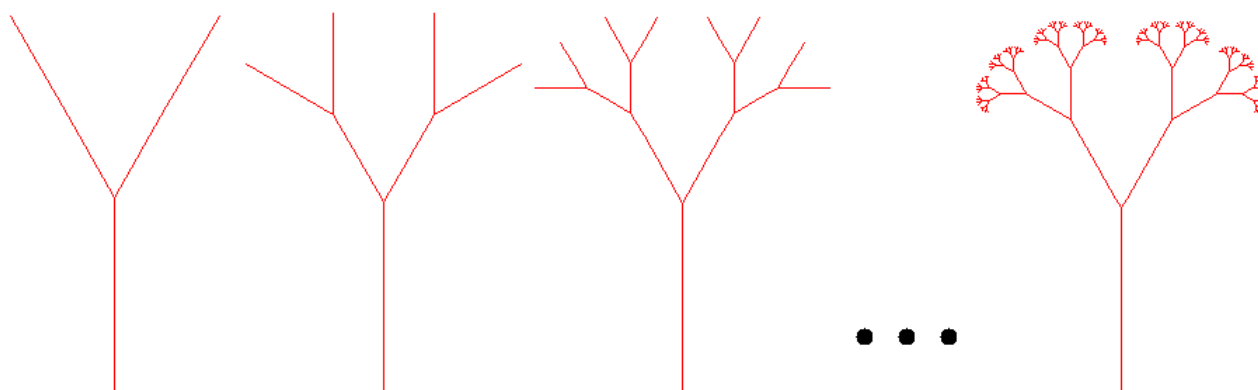
popsány maďarským matematikem a biologem Aristidem Lindenmeyerem v roce 1968, který je využíval pro simulaci růstu rostlin avšak brzy přišel na to, že za pomoci jejich gramatiky lze také konstruovat veškeré typy soběpodobných fraktálů.



Obrázek 2. Přírodní fraktální útvary

## 4.2 Soběpodobný fraktál

Ačkoliv téměř všechny fraktály mají napohled velice složitý tvar, mnoho je jich tvořeno kopiemi stejných částí pouze s jiným měřítkem. U dokonalé soběpodobnosti platí, že ať pozorujeme soběpodobný objekt v jakémkoliv měřítku nebo v jakémkoliv rozlišení, pozorujeme stále se opakující charakteristický tvar. Typickými příklady soběpodobných objektů můžeme opět nalézt v přírodě – například zmíněné kapradí, mraky, hory, stromy...



Obrázek 3. Soběpodobnost fraktálních obrazců

### **4.3 Praktické využití fraktálních obrazců**

Fraktální obrazce našly od doby svého objevu panem Mandelbortem již mnoho možností využití ve vědě a technice. Jsou využívány vědci od různých teoretických simulačních geometrických výpočtů až po praktické využití zejména v moderní elektrotechnice, biologii, biochemice a medicíně. Například již zmíněná elektrotechnika našla využití fraktálů a zejména L-Systemů v praktickém vynálezu takzvané fraktálové antény, jež je v současné době díky své malé velikosti a kompaktnosti a přitom širokým pásmovým rozsahem, používaná především armádou. Dalším možným využitím L-Systemů v elektrotechnice je nalezení vhodného rozmístění součástek na plošném spoji, což bývá zejména u součástek z velkým počtem vývodů (např. mikroprocesory) obvykle problém.

### **4.4 Princip funkce L-Systemů**

L-Systemy jsou rekurzivní přepisovací systémy. Formálně je můžeme definovat jako trojici  $G = (\Sigma, R, A)$  kde  $\Sigma$  je abeceda tedy konečná neprázdná množina znaků,  $R$  je množina přepisovacích pravidel a  $A$  (dále axiom) je výchozí řetězec ze kterého celý systém vychází, vzniká. Celý princip spočívá v postupné iteraci a tím postupným paralelním nahrazováním prvků z množiny  $\Sigma$  dle daných přepisovacích pravidel. Tím daný L-System nabývá na komplexnosti a stává se čím dál tím více detailnějším. Konec iterace nastává kdy již na současný systém nelze aplikovat žádné pravidlo z množiny  $R$ . Je zřejmé že tato situace také nemusí v některých případech vůbec nastat, proto se tedy iterace provádí po postupných krocích. Tím také můžeme pozorovat postupný vývoj (růst) L-Systemu.

## 4.5 Přepisovací pravidla a řídicí znaky

Pro vykreslení jakéhokoli L-Systému se používá technika známá jako želví grafika. Jedná se o způsob vykreslování pomocí imaginární želvy, jež jako vstup dostane řetězec s příkazy které následně vykonává. Příkazů může být mnoho, záleží na gramatice daného vykreslovacího enginu. Já jsem zvolil gramatiku vlastní, s cílem implementovat mnoho příkazů pro pohyb želvy. Výčet veškerých příkazů a jejich bližší specifikace spolu z jejich funkcí je uveden v tabulce.

Znak	Význam
+	Otočí želvu o úhel $\alpha$
-	Otočí želvu o úhel $-\beta$
[	Uloží aktuální pozici a směr želvy na zásobník
]	Vybere naposledy přidanou pozici a směr želvy ze zásobníku
?	Otočí želvu o náhodný úhel
	Otočí želvu o 180

Tato gramatika je dále rozšířena o znaky pro pohyb želvy. Znaky pro pohyb želvy definuje uživatel, může použít celou abecedu (jsou rozlišována velká a malá písmena). Samotná želva pak akceptuje čtyři příkazy pro pohyb a to – krok dopředu nebo dozadu s kreslením čáry, krok dopředu bez vykreslení čáry a stání na místě. Mohlo by se zdát, že příkaz pro stání na místě je zbytečný, ale používá se jako terminální znak důležitý pro rozvoj fraktálu. Podrobněji je tato problematika popsána v kapitole 6.7.

## 5 Implementace

### 5.1 Vývojová platforma a prostředí

Jelikož jsem zastáncem otevřených (tj. opensource) řešení a jsem také fanouškem multiplatformních aplikací, čekal mě nelehký úkol a to vybrat takovou platformu pro vývoj aplikace která by ji zajistila multiplatformnost, lehkost implementace a umožnila projekt prezentovat jako opensource. Na užší výběr jsem vzal v potaz 2 vývojové platformy které všechno toto víceméně nabízely a to C++ a Java™. Hlavní výhody implementace v C++ jsou určitě vyšší rychlost výsledného binárního kódu, avšak pro C++ neexistuje žádný framework pro komplexní vývoj aplikace, který by ji poskytl takové výhody jaké nabízejí defaultní třídy v jazyce Java. Proto jsem se po důsledném uvážení rozhodl pro platformu Java™. Jako vývojový nástroj jsem zvolil NetBeans IDE (<http://netbeans.org/>) které poskytuje komfortní a obsáhlé prostředí pro vývoj aplikací nejen v jazyce Java. Celý vývoj probíhá na operačním systému GNU/Linux konkrétně distribuci xubuntu spolu s občasným testováním na operačních systémech Mac OS X a Microsoft Windows Vista.

## **5.2 Požadavky na GUI programu**

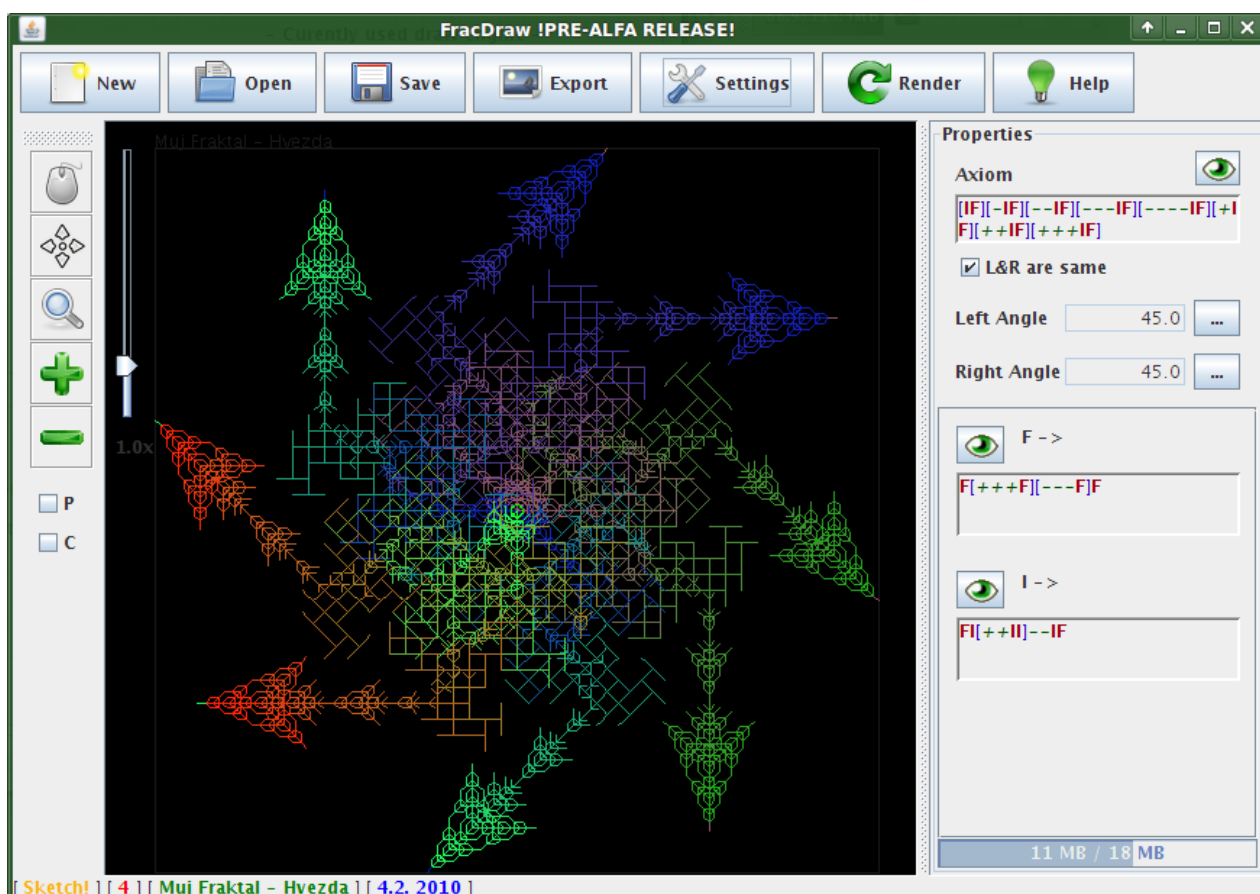
Aplikaci jsem již od samého začátku chtěl koncipovat pro použití v grafickém uživatelském prostředí, tak jak je dnes mezi uživateli počítačů zvykem. Důležitým faktorem pak také bylo, jaké rozvržení grafických komponent zvolit. Protože jsem chtěl využít co největší prostor pro zobrazení fraktálu, rozhodl jsem se pro systém postraních nabídek, a největší prostor uprostřed obrazovky nechat na zobrazení vykresleného fraktálu. Vzhledem k tomu že jedním z mých hlavních požadavků na GUI byla jeho uživatelská přívětivost, rozhodl jsem se nechat na hlavním formuláři co nejméně grafických komponent které poskytují základní operace s fraktálem a pokročilejší funkce a nastavení jsou přístupné v podformulářích. Také jsem vypustil použití programového menu, jelikož bloudění v „džungli“ nabídek mi jako uživatelsky přívětivé příliš nepřijde a ovládacích tlačítek které uživatel potřebuje používat často v programu tolik není. Tento trend lze již například vysledovat u moderních prohlížečů webu a osobně myslím že se postupem času bude také od menu i v jiných „profesionálnějších“ programech upouštět.

## **5.3 Požadavek na grafický návrh předpisu L-Systému**

Další myšlenkou, jež program provázela již od fáze návrhu před započítím implementace byla na grafický návrh předpisu L-Systému. Jedná se o dle mého názoru naprosto unikátní funkci programu, kdy místo obvyklého ručního zadávání pohybových příkazů, má uživatel možnost tyto příkazy interaktivně „naklikat“ v grafickém režimu. Ani po důkladném hledání v hlubinách internetu jsem nenašel program s podobnou funkcionalitou, a tak by se dalo říci že se jedná o naprosto jedinečnou funkci na poli editorů fraktálních obrazců.

## 6 Základní GUI programu

Během vývoje programu se základní GUI změnilo několikrát, zejména kvůli technickým problémům s implementací nebo díky neefektivitě či uživatelské „nepříjemnosti“ daného řešení, avšak základní návrh jsem měl vypracovaný již od počátku své práce.



Obrázek 4. Konečný vzhled hlavního okna programu

Celá hlavní plocha (uprostřed) programu je věnována pro vykreslení aktuálně rozpracovaného fraktálu. Vrchní panel obsahuje tlačítka s nejčastěji využívanými akcemi při zpracovávání fraktálu. Veškeré dodatečné nastavení je pak dostupné ve formuláři který lze vyvolat tlačítkem „Settings“. Levá strana obsahuje toolbar s tlačítky pro změnu zobrazení fraktálu. Pravá strana pak obsahuje komplexní panely ve kterých může uživatel nastavit či si prohlédnout veškerá přepisovací pravidla a nastavení úhlů. Ve spodní části tohoto pravého panelu se nachází ukazatel právě obsazené (případně volné) paměti RAM. V úplně spodní části programu se pak nachází statusbar.



## 6.1 Vykreslovací plocha

Vzhledem k tomu, že uživatel v programu tvoří práci jejíž hlavním výsledkem je grafický výstup, můžeme uvažovat že tato komponenta je používána v programu opravdu často. Uživatel zde vidí dosavadní výsledek své práce, a také to jak bude vypadat fraktál, když jej vyexportuje do jiného, například rastrového formátu. Práce s touto komponentou je částečně obstarána toolbarem který je po pravé straně v defaultním nastavení. Základní akce které umožňuje tato komponenta zahrnují drag&drop se současným fraktálem, změnu počátečního bodu vykreslování a změnu zoomu (tj. Přiblížení). Pro změnu zoomu můžeme použít několik způsobů. Prvním z nich je použití tlačítka Zoom z toolbaru. Druhým způsobem je použití kolečka myši pro přiblížení/oddálení fraktálu. K tomuto účelu slouží také tzv. Zoom panel který se nachází ve vykreslovací ploše. Tento panel také zobrazuje aktuální faktor přiblížení.



Obrázek 5. Zoom panel

## 6.2 Panel s vlastnostmi fraktálu

Na tomto panelu se nachází přehled všech prepisovacích pravidel, včetně definice axiomu a nastavení úhlů. Vzhledem k možnostem tohoto programu je možné nastavit levý a pravý úhel zvlášť, což je mimochodem možnost, která je mnohdy různými konkurenčními programy opomíjena, i přestože se jedná o



Obrázek 6. Panel vlastností

velmi užitečnou vlastnost. Chce-li přesto uživatel mít definován pouze jeden úhel, stačí zaškrtnout políčko s informací že levý a pravý úhel jsou stejné a změna jakéhokoliv z obou úhlů se projeví i na úhlu druhém. Opět se jedná o šikovnou pomůcku, která však rozhodně nebývá standardem.

Lze si také všimnout, že u každého prepisovacího pravidla a axiomu je také uvedeno jeho plné znění, kteréžto je navíc přehledně a barevně obarveno pro lepší přehlednost a pochopení. Každé toto pravidlo obsahuje také tlačítko pro jeho zobrazení a úpravu. Ta je dle mého názoru řešena naprosto neobvyklou, a dalo by se říci i revoluční metodou. Více o tomto v kapitole 6.5 o editačním okně.

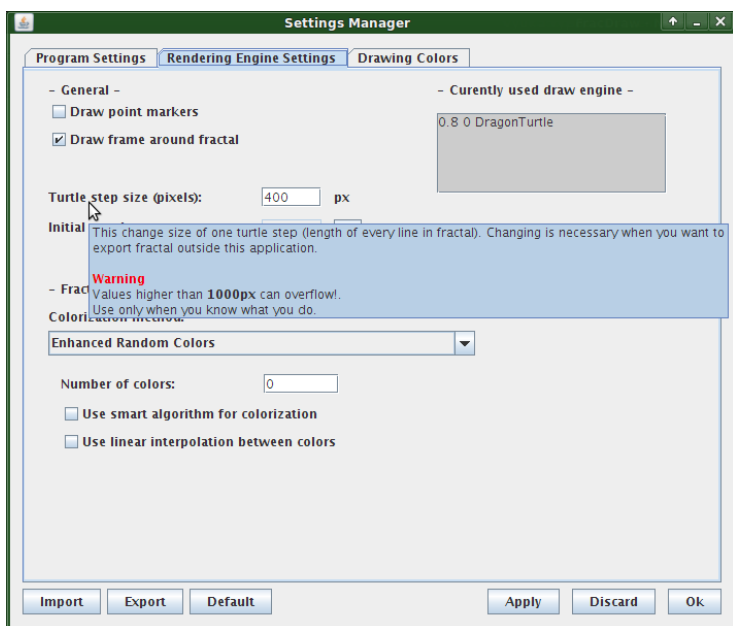
Úplně ve spodní části je pak zobrazeno využití operační paměti RAM. Vykreslování fraktálů je totiž časově i paměťově náročná operace a zejména u konkurenčních programů zabývajících se touto tematikou může lehce dojít k zamrznutí programu z důvodu nedostatku paměti. Můj software se snaží tento problém řešit pomocí odhadu o náročnosti výpočtu před zahájením další iterace. Tento odhad je založen především na délce výsledného řetězce pro danou iteraci. Avšak ani tento způsob vzhledem k tomu že se jedná o pouhý odhad nemusí uživatele zachránit před potenciálním

zamrznutím programu. Tímto si tedy uživatel může vizuálně ověřit že mu bude jeho operační paměť stačit i pro další výpočet.

### 6.3 Dialog „Správce nastavení“

Správa nastavení patří u mnoha programů k nejkompexnějším a nejvíce detailnějším nástrojům, jelikož slouží k nastavení možností a vlastností všelijakých funkcí daného programu. Nejinak je to i u tohoto software, což je zapříčiněno také tím že zejména u vykreslovacího enginu je možných vlastností které výrazně mění parametry vykresleného obrazce opravdu velmi mnoho. Avšak i pokud je nastavení obsáhle, neznamená to nutně že takovýto správce nastavení musí být nepřehledný a uživatelsky doslova nepřátelský. Proto je u tohoto software dialog určený k nastavení parametřů koncipovaný zejména pro jednoduché použití a lehkou orientaci v něm. Všechny možnosti jsou adekvátně setříděny v příslušných kategoriích a pro každou možnost je k dispozici malá

avšak informacemi oplývající nápověda, která se objeví po najetí myši na danou nabídku, jak je ostatně vidět na obrázku číslo 7.



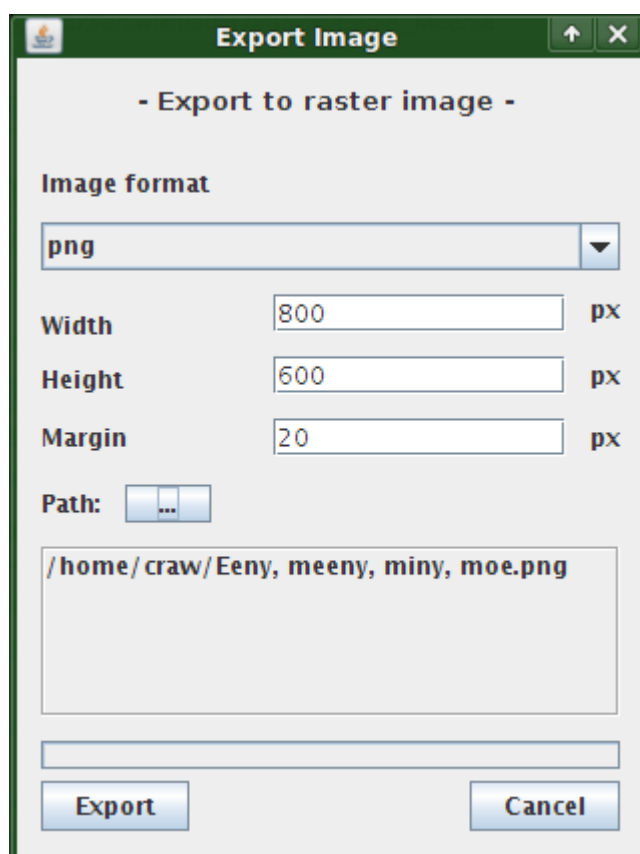
Obrázek 7. Správce nastavení

Ovšem nejen přehledností a jednouchostí musí oslňovat kvalitní správce nastavení. Všechno je uživateli k ničemu pokud neexistuje pořádná správa už jednou nastavených možností. To tento správce řeší

pomocí importace a exportace nastavení do souboru. Výchozí nastavení je možno obnovit pomocí tlačítka Default.

## 6.4 Dialogy „Export“

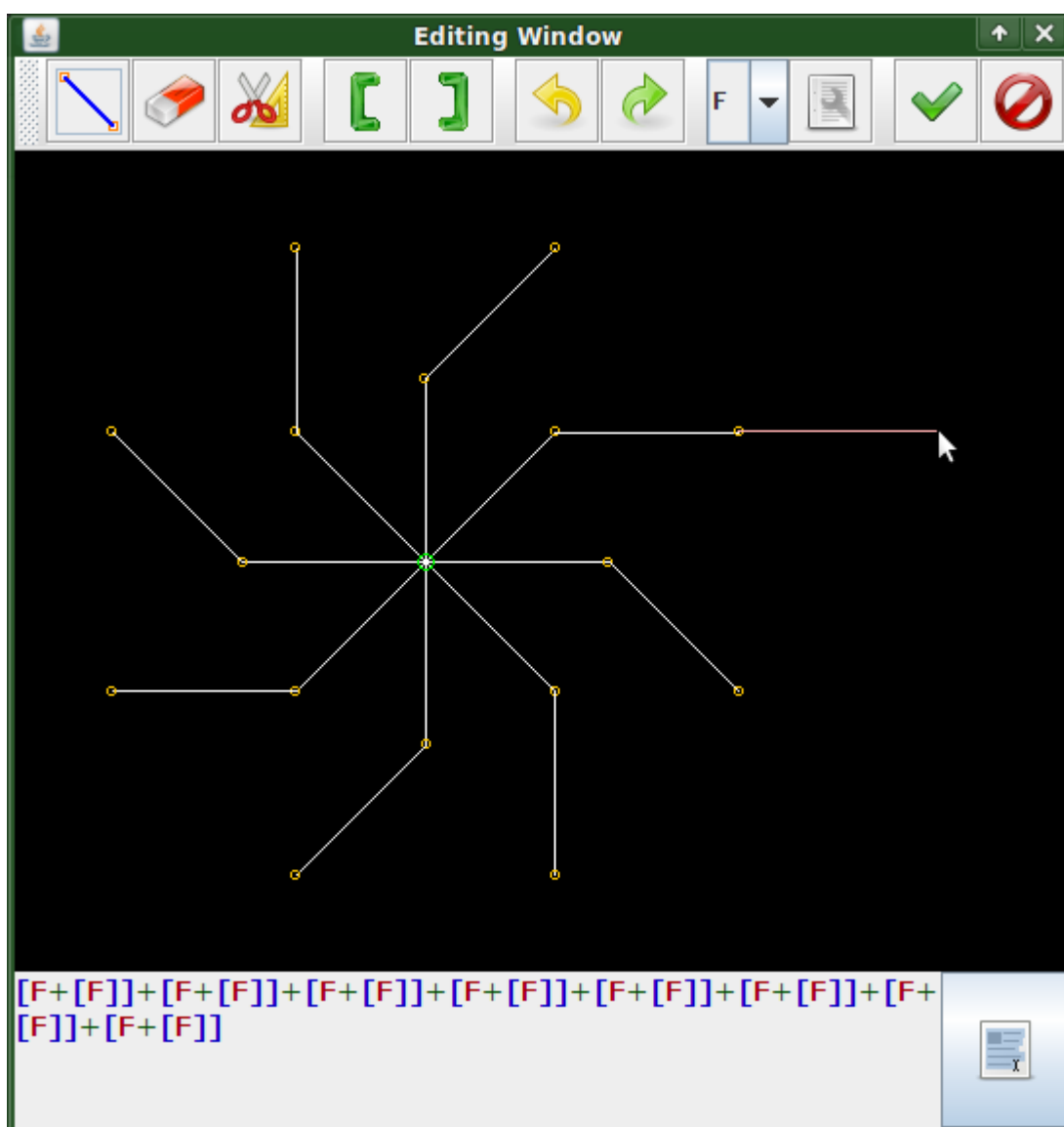
Výsledkem programu je jednak textový a jednak grafický výstup. I když připouštím že grafický výstup není zrovna nejsilnější stránkou tohoto programu (a ani nemá být, program je totiž spíše zaměřen na návrh než na vykreslování), aplikace přeci jen nabízí základní funkcionalitu pro výstup výsledného fraktálu jednak jako rastr a jednak i jako vektor. Na obrázku 8 je zobrazena podoba dialogu pro export rastrového obrázku. Pro rastrový výstup je možno použít jakýkoliv formát jež podporuje Java™ na dané platformě, minimálně jsou však dostupny tyto formáty: GIF, JPEG, PNG (na platformě windows pak ještě BMP). Jako vektorový výstupní formát jsem zvolil otevřený formát vektorové grafiky SVG.



Obrázek 8. Export do rastru

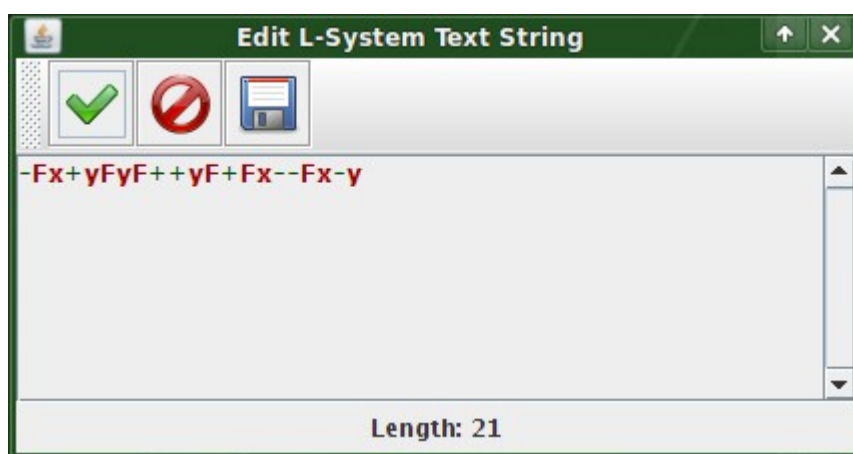
## 6.5 Dialog „Editace“

Tento dialog slouží k editaci přepisovacího pravidla, nebo axiomu. Uživatel má 2 možnosti, jak editovat předpis – buď může použít tlačítko na editaci řetězce, které se nachází vedle barevně vypsání řetězce, anebo může použít zmíněný unikátní interaktivní klikací systém. Celé klikací rozhraní funguje jako velmi jednoduchý vektorový editor. Uživatel má k dispozici několik nástrojů, které jsou uspořádány v toolbaru, kterým lze v programu různě přesouvat. Kromě „kreslítka“ a gumy má uživatel k dispozici také ovladače zásobníků želvy, nástroj pro krok zpět a nastavení pohybových pravidel želvy.



Obrázek 9. Dialog Editace

Při každém kliknutí dojde k doskreslení právě přidaného kroku a také k aktualizaci vypisovaného řetězce. Přestože je grafický návrh L-Systému opravdu rychlou a elegantní cestou, uživatel v některých případech zjistí že je mnohem výhodnější použít ruční editace řetězce (zejména, pokud potřebujeme upravit část jež se nenachází na konci řetězce). Program samozřejmě komfortní nástroj pro ruční editaci nabízí. Jeho prostředí můžeme vidět na obrázku 10.

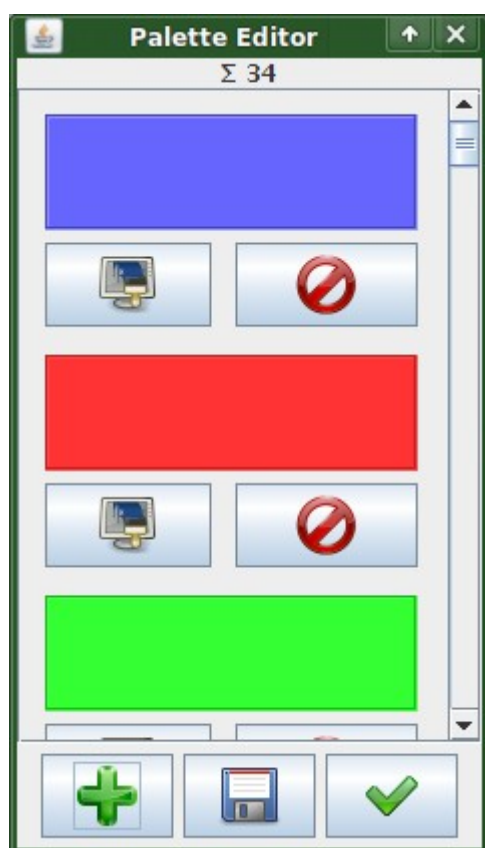


Obrázek 10. Dialog Editace textového řetězce

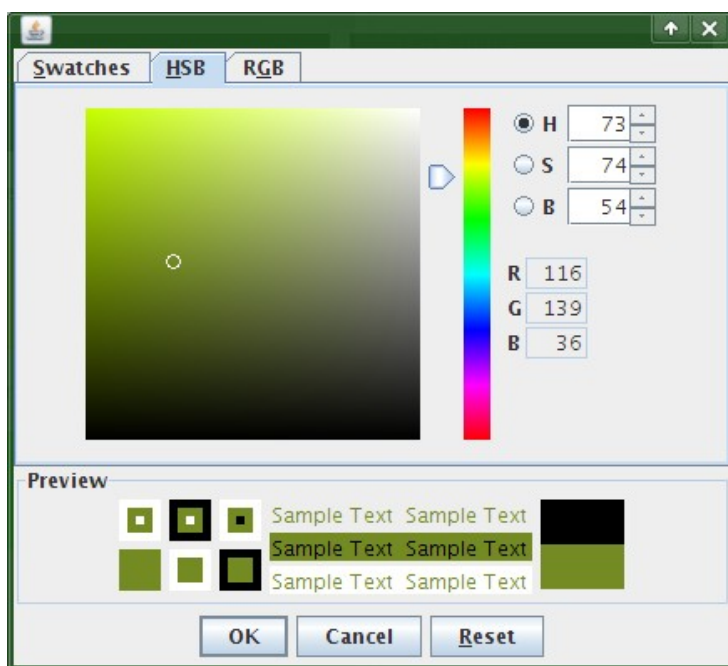
Tuto utilitku může uživatel taktéž použít pro prohlížení a/nebo uložení vygenerovaného řetězce při určité iteraci právě vykreslovaného fraktálního obrazce. Uložení výsledného řetězce do souboru je vhodné zejména tehdy, když chceme řetězec vygenerovaný tímto programem vykreslit v jiném programu, který například umí pouze želví grafiku.

## 6.6 Dialog „Editor palety“

Při obarvování fraktálu je možno využít několik metod, které mimo jiné zahrnují i obarvení fraktálu pomocí palety. To a fakt, že jsem se nerozhodl použít žádný z formátů které jsou používány k uchovávání barevných palet zejména díky jejich obsáhlosti a množství uchovávaných hodnot, které by stejně zůstaly nevyužity, mě vedlo k vytvoření minimalistického editoru barevných palet. Editor nabízí uživateli pouze základní operace a to přidávání a odebírání barvy z palety, změnu jednotlivých přidaných barev v paletě. Také je možno výslednou paletu uložit do souboru, nebo naopak načíst ze souboru paletu určenou k editaci. Formát souboru palety je uveden v příloze.



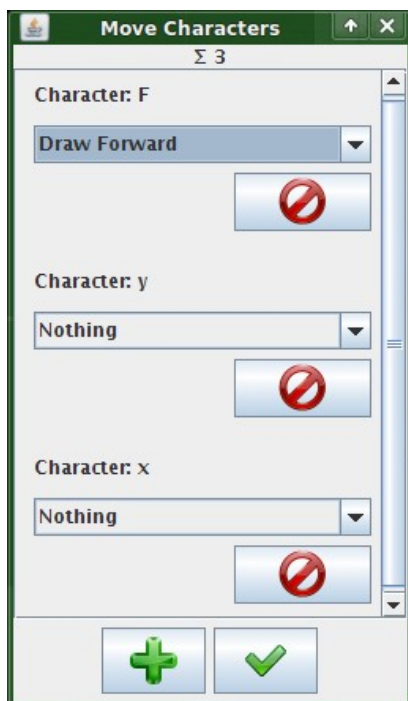
Obrázek 11. Editor Palety



Obrázek 12. Dialog výběru barvy poskytovaný Java API

## 6.7 Dialog „Editor pohybových pravidel“

Důležitým faktorem při vykreslování L-Systemu je také to, jak budou interpretovány znaky pro pohyb virtuální želvy. Znaky u kterých se předem ví, jakou činnost vykonávají, jsou uvedeny v tabulce 1. Veškeré ostatní znaky jsou



Obrázek 13. Editor znaků

možnosti jeho nastavení:

- **Vpřed s kreslením čáry**

Nejpoužívanější režim. Želva učiní jeden krok dopředu (tzn. nakreslení čáry).

- **Vzad s kreslením čáry**

Želva učiní krok dozadu s kreslením čáry. Vhodné zejména pokud je příliš malý úhel otočení, nebo úhel otočení není násobkem 180°.

- **Stání**

Želva stojí na místě. Znamé také jako terminální operátor.

- **Vpřed bez kreslení čáry**

Želva učiní jeden krok dopředu bez kreslení čáry.

definovány uživatelem, a uloženy spolu s předpisem L-Systemu do souboru. V jiných software je problematika řešena často jedním z těchto způsobů – buď je definována pevná abeceda znaků s danou činností anebo se předpokládá pro každý znak stejná činnost (pohyb vpřed s kreslením čáry). Jenom opravdu malé procento software s podobným zaměřením umožňuje nechat uživatele definovat si symboly dle svého uvážení nebo potřeby. Jelikož se tato práce snažila obsáhnout problematiku řádně, je pro uživatele možné definovat si vlastní kolekci znaků a jejich významy. Pokud tedy želva narazí na uživatelsky definovaný znak jsou možné tyto čtyři



Uživatelský znak může být jakýkoliv znak z malé nebo velké abecedy. Velikost písmen je striktně rozlišována, malé 'a' je tedy bráno jako jiný znak než velké 'A'. Nastavení znaků je realizováno pomocí dialogu.

## **6.8 Průvodce vytvořením nového L-Systemu**

Aplikace umožňuje samozřejmě mimo možnosti úpravy již hotových předpisů i návrh předpisů nových. Toto je realizováno pomocí grafického průvodce vytvořením nového L-Systemu. Ten se v několika krocích zeptá na veškeré základní parametry vytvářeného L-Systemu (úhly, jméno, pohybové znaky, axiom, předpisová pravidla). Nově navržený fraktál poté nabídne uložit, protože jak vyplývá z praxe, výpadek proudu a s tím související ztráta nově vytvořených dat není něco co pracujícího zrovna potěší.

## 6.9 Statusbar

Statusbar má v této aplikaci velkou informační hodnotu, jelikož je to místo kde se uživatel dozví spoustu informací o právě probíhajících operacích. Statusbar je také uživatelem plně konfigurovatelný – to vede k tomu, že uživatel kterému nepřipadají informace jím zobrazované přehledné, či jsou na jeho platformě nečitelné či nějak jinak deformované, může jejich umístění, barvu nebo styl zobrazení nakonfigurovat dle libosti. U celé aplikace se musí brát v potaz to, že je multiplatformní a může tedy běžet na rozličných operačních systémech a zdaleka ne v každém je šedé pozadí standardem tak jako v operačním systému firmy Microsoft. Jelikož většina uživatelského prostředí používá ke komunikaci s uživatelem ikonky, byl statusbar, který komunikuje s uživatelem výhradně pomocí textových informací nejnáklonějším na změny nastavení na různých platformách. Toto je vyřešeno pomocí jednoduché syntaxe s níž lze nakonfigurovat status bar naprosto dle představ uživatele. Celý status bar lze nakonfigurovat pomocí jednoho řádku textu. Tento řádek obsahuje veškeré informace které chceme na statusbaru zobrazit, oddělené středníkem. Každé toto políčko pak může být rozděleno pomocí znaku '|' (roura). Před rourou pak následuje definice stylu písma v tomto políčku.

Zatím podporované znaky jsou:

Znak	Význam
<b>r</b>	změna písma na červenou barvu
<b>g</b>	změna písma na zelenou barvu
<b>b</b>	změna písma na modrou barvu
<b>y</b>	změna písma na žlutou barvu
<b>s</b>	změna písma na tučné
<b>u</b>	změna písma na podtržené


Na pravé straně (za rourou) pak následuje text, který má být zobrazen v tomto políčku. Může obsahovat několik proměnných které jsou přepsány hodnotou nebo vlastností již mají vyjadřovat. Každá proměnná pro přehlednost obsahuje na začátku svého názvu znak procenta. Podporované jsou tyto proměnné:

Proměnná	Význam
<b>%fractal</b>	Jméno fraktálu
<b>%length</b>	Délka řetězce právě vykresleného fraktálu
<b>%iteration</b>	Číslo současné iterace
<b>%date</b>	Dnešní datum

Ukázka použití:

#### Výchozí řetězec

"ysu|FracDraw!;r|%iteration;gs|%fractal;ys|L-System length: %length;b|Date: %date"

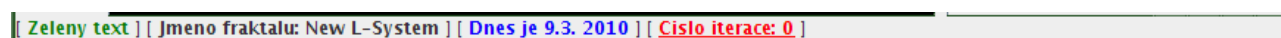


[FracDraw!][5][Rostlina 3][L-System length: 570][Date: 9.3. 2010]

Obrázek 14. Výchozí statusbar

#### Upravený řetězec

"g|Zeleny text;Jmeno fraktalu: %fractal;bs|Dnes je %date;ur|Cislo iterace: %iteration"



[Zeleny text][Jmeno fraktalu: New L-System][Dnes je 9.3. 2010][Cislo iterace: 0]

Obrázek 15. Příklad uživatelsky upraveného statusbaru

## **7 Závěr**

### **7.1 Souhrn**

Cílem projektu bylo vytvoření aplikace jež by splňovala jak základní, tak pokročilé požadavky na software použitý při tvorbě předpisů různorodých L-Systemů, úkolem jež mluví sám za sebe pak bylo vytvoření editoru schopného graficky navrhnout předpis L-Systemu. Cíl práce se dle mého názoru podařilo uspokojivě splnit.

### **7.2 Budoucnost projektu**

Hlavním záměrem a inspirací tohoto projektu bylo úspěšné splnění dlouhodobé maturitní zkoušky, případně jiných soutěží jako je například SOČ. Po finálních úpravách bude program uvolněn pod opensource licencí GNU/GPL verze 3, jež zajistí každému zájemci dostupnost zdrojových kódů pro potřebu vlastního vývoje, výuky návrhu a zpracování fraktálních obrazců a zejména L-Systemů, případně výuky programovacích technik jež byly při implementaci tohoto software použity. Uvolnění pod opensource licencí tak skýta aplikaci několik dalších možností využití, které prakticky ani nesouvisejí s technickým zaměřením aplikace.

## 8 Seznam použitých zdrojů

TIŠNOVSKÝ, Pavel. *L-Systémy: Přírodní objekty i umělé artefakty* – Root.cz [online], 2006-24-10. Dostupné z: <<http://www.root.cz/clanky/l-systemy-prirodni-objekty-i-umele-artefakty/>>

TIŠNOVSKÝ, Pavel. *Implementace L-systémů založená na želví grafice* – Root.cz [online], 2006-31-10. Dostupné z: <<http://www.root.cz/clanky/implementace-l-systemu-zalozena-na-zelvi-grafice/>>

*Fraktál* – wikipedia.org [online]. Dostupné z: <<http://cs.wikipedia.org/wiki/Fraktál>>

*Fractal* – wikipedia.org [online], anglicky. Dostupné z: <<http://en.wikipedia.org/wiki/Fractal>>

*L-System* – wikipedia.org [online], anglicky. Dostupné z: <<http://en.wikipedia.org/wiki/L-System>>

*Iterated function system* – wikipedia.org [online], anglicky. Dostupné z: <[http://en.wikipedia.org/wiki/Iterated\\_function\\_system](http://en.wikipedia.org/wiki/Iterated_function_system)>

## 9 Použitý software

**Netbeans IDE** - <<http://netbeans.org/>>

*Vývojové prostředí celého projektu.*

**Sun Java JDK** - <<http://java.sun.com/>>

*Sada kompilátorů, interpretu a ladících prostředků pro vývoj v programovacím jazyce Java.*

**GNU/Linux** - <<http://www.linux.cz/>>

*Hlavní vývojová platforma celého projektu.*

**OpenOffice.org** - <<http://www.openoffice.org/>>

*Sada kancelářského software použitá k vytvoření této dokumentace.*

**GIMP** - <<http://www.gimp.org/>>

*Editor rastrové grafiky použitý k úpravě obrázků.*

**Inkscape** - <<http://www.inkscape.org/>>

*Editor vektorové grafiky použitý k úpravě obrázků a kontrole korektnosti vygenerovaného SVG kódu.*

**Mercurial** - <<http://mercurial.selenic.com/>>

*Distribuovaný systém zprávy verzí použitý při vývoji k udržení přehlednosti změn ve zdrojovém kódu.*