

SFC – Software Computing - Projekt 2017

Vizualizácia Juliových množín

Michal Gabonay (xgabon00@stud.fit.vutbr.cz)

Brno 11.11.2017

1. Úvod

Tento dokument popisuje riešenie projektu do predmetu Soft computing na FIT VUT v Brne. Úlohou projektu bolo vytvorenie programu, ktorý bude vykresľovať fraktál tvorený Juliovou množinou. Postupne popíšeme čo sú to fraktály a Juliove množiny, ich vlastnosti a princípy. V poslednej časti sa budeme zaoberať samotnou realizáciou aplikácie, algoritmom a spôsobom ako túto Juliovu množinu vykresľujem vo svojej aplikácii a stručný popis ovládania aplikácie obohatený a názorné ukážky fungovania aplikácie.

2. Fraktály a Juliova množina

Fraktál je geometrický objekt vybudovaný pomocou rekurzcie. Základná vlastnosť fraktálov je sebedobnosť. **Sebedobnosť** znamená, že ak daný nepravidelný, fragmentovaný geometrický tvar pozorujeme v akomkoľvek merítke, tak sú viditeľné, stále sa opakujúce časti a charakteristické tvary. V podstate približne podobné, zmenšené kópie celého geometrického tvaru. Fraktály majú zväčša veľmi zložité geometrické tvary, ktoré sú však generované opakovaným použitím jednoduchých pravidiel. Najznámejším fraktálom je Mandelbrotova množina, ktorá nesie meno po človeku, ktorý ako prvý použil pojem fraktál. V tomto projekte sa však zameriame na ďalší známy fraktál, ktorý je tvorený pomocou Juliovej množiny.

Juliova množina je množina všetkých bodov z v komplexnej rovine, pre ktorý platí, že vzťah $z_{n+1} = z_n^2 + c$ nekonverguje. Symbol c reprezentuje ľubovoľné komplexné číslo. Na prvý pohľad je vidno, že Juliovu množinu charakterizuje práve toto komplexné číslo. Hranica tejto množiny tvorí vyššie spomínaný útvar, fraktál.

3. Programová vizualizácia

Pre vytvorenie aplikácie na vykresľovanie Julioových množín som si vybral programovací jazyk Java. Výsledná aplikácia je vytvorená a zobrazená konkrétne pomocou JavyFx.

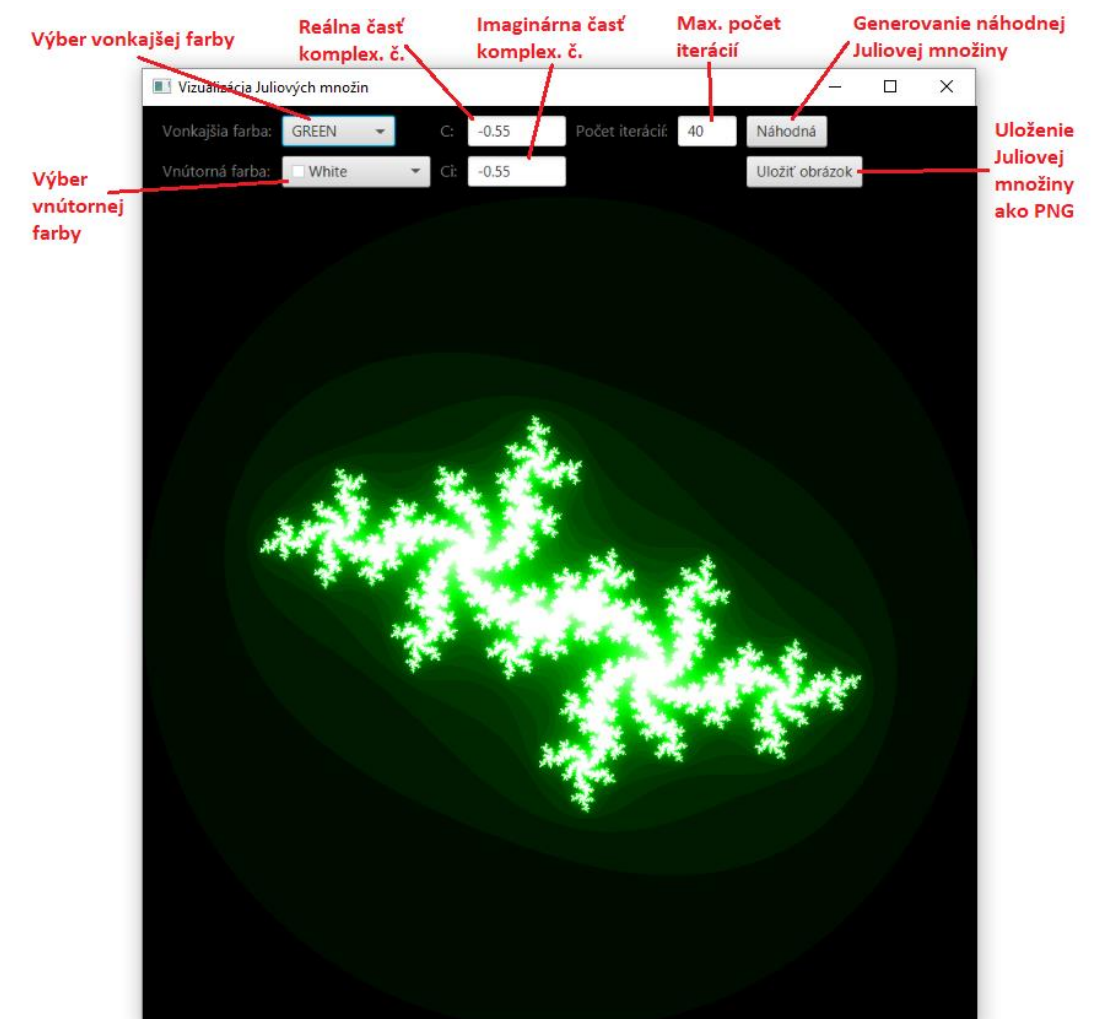
Keďže využívam množstvo premenných pre popis Juliovej množiny, rozhodol som sa pre použitie JavaBeans. Nejde o nič prevratné, je to len štandard pre implementáciu. V podstate je to trieda zo všetkými premennými privátnymi a obsahujúca pre všetko potrebné getter a setter. Táto konvencia značne uľahčuje ďalšiu prácu. Aplikácia začína práve inicializáciou tohto JavaBean pre Juliovu množinu (vloženie prvotných hodnôt pre jej vykreslenie). Nasleduje inicializácia a vytvorenie JavaFx aplikačných objektov pre zobrazovanie aplikácie a manipulovanie s ňou. Tým sa myslia prvky na ovládanie aplikácie, ktorý podrobnejšie popíšeme na konci tejto kapitoly. Nasleduje samotné vykreslenie Juliovej množiny.

Algoritmus pre samotný výpočet a vykresľovanie Julioových množín nie je vôbec zložitý. Celý sa skladá z niekoľkých krokov, ktoré si popíšeme. Prechádzam celé okno kde sa daná množina má vykresľovať (vopred známe, fixne zadané) pixel po pixeli. Takto pre každý pixel, na základe jeho súradníc (x -ová súradnica je imaginárna časť, y -ová súradnica je reálna časť komplexného čísla) vypočítam, či sa daný bod (pixel) nachádza v Juliovej množine alebo nie. Výpočet je založený na vyššie spomínanom vzťahu pre platnosť

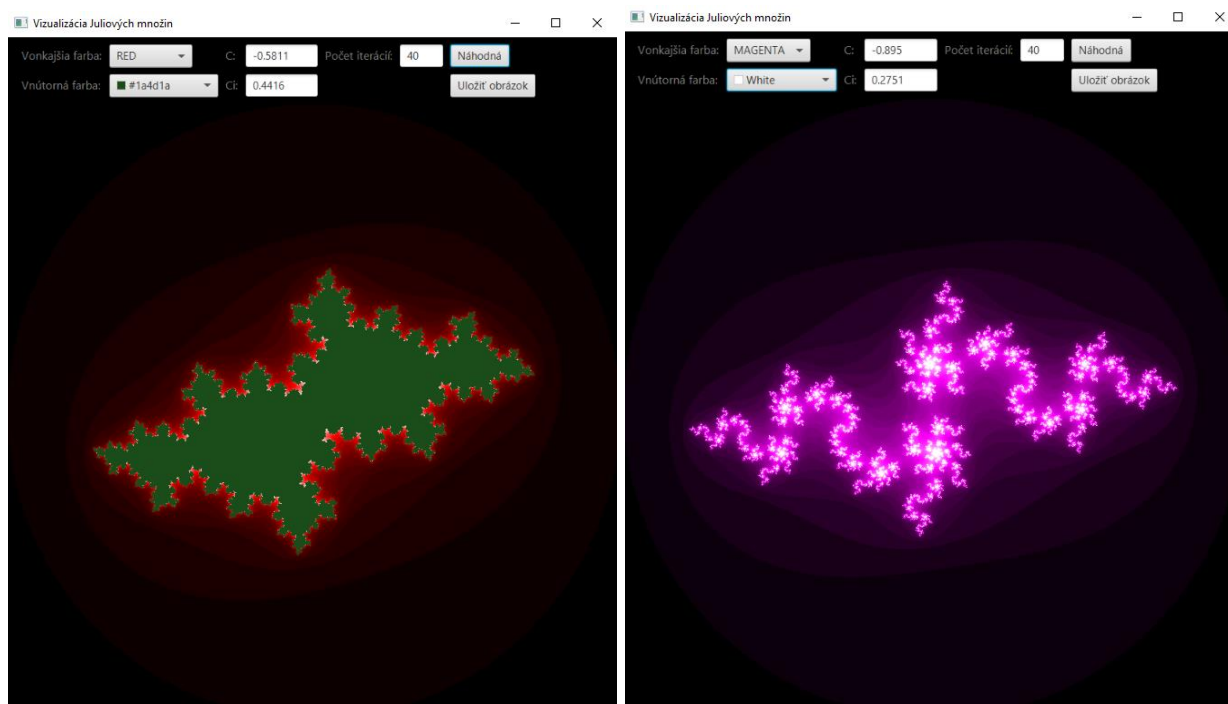
Juliových množín. Výpočet prebieha postupne po iteráciách, zisťujem či absolútna hodnota bodu (súradnice skúšaného pixelu - komplexné číslo) je väčšia ako 2. Ak nie prirátam k nemu hodnotu C (komplexné číslo charakterizujúce J . množinu) a iterujem ďalej. Ak podmienka neplatí preruším výpočet a vraciam, v ktorej iterácii podmienka prestala platiť. Na základe toho určujem sýtosť vonkajšej farby. Bod patrí do Juliovej množiny, ak výpočet vráti zadaný maximálny počet iterácií.

Pre grafické zobrazovanie rozlišuje dva typy farieb a to pre obsah Juliovej množiny (vnútorná farba) a pre body čo nepatria do výslednej množiny, ale sú pozostatkom z predošlých iterácií (vonkajšia farba). Vnútorná farba je riešená pomocou JavaFx ColorPicker. Pre vonkajšiu farbu je možnosť vybrať z viacerých schém. Vybranej farby sýtosť, ako už bolo vyššie spomínané, závisí od maximálneho počtu iterácií a čísla iterácie, kedy bolo zistené, že daný bod už do množiny nepatrí. Princíp je založený na RGB zložkách farieb. Podľa vybranej farebnej schémy buď danú RGB zložku vyzdvihnem alebo potlačujem.

4. Ukážka aplikácie a popis ovládania



Obr. 1 – Príklad aplikácie s popisom ovládacích prvkov



Obr. 2 – Príklad výstupu aplikácie

Ovládanie aplikácie je pomerne dosť intuitívne. Vonkajšia farba sa vyberá pomocou select boxu, vnútorná farba sa vyberá z palety farieb, komplexné číslo sa wpisuje do textového poľa “C”, pre reálnu časť a “Ci”, pre imaginárnu časť. Potvrďuje sa enterom, reálna aj imaginárna časť. Generovanie náhodnej Juliovej množiny je jednoducho pomocou stlačenia tlačidla “Náhodná” a vytvorenú Juliovu množinu je možné uložiť ako PNG obrázok.

5. Záver

Cieľom projektu bolo vytvoriť aplikáciu pre vizualizáciu Juliových množín. V tomto dokumente sa popisuje, čo vlastne sú tie Juliove množiny a hlavne moje riešenie a spôsob realizácie tejto problematiky. Aplikácia môže slúžiť na vytváranie názorných ukážok Juliových množín, ktoré s trochou úsilia vedia byť veľmi pekné po grafickej stránke.

Zdroje

- Prezentácie z predmetu SFC
- *Fraktály v počítačové grafice*. [Online; navštíveno 14.11.2017].
URL <https://www.root.cz/clanky/fraktaly-v-pocitacove-grafice-x>