

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
KIV/KPG

Fraktály

Pavel Zelenka
A16B0176P
zelenkap@students.zcu.cz

29. dubna 2018

1 Zadání

Zadáním úkolu je vytvoření nového programu nebo rozšíření programu ze cvičení, tak aby vykresloval fraktály, které nebyly na cvičení.

2 Analýza problému

Fraktál je sebedpodobný útvar, to znamená, že lze pozorovat stále opakující se tvar. Mezi známe fraktály patří Sierpińského koberec, Sierpińského trojúhelník, či křivka vyplňující prostor.

V této práci se budu zabývat křivka vyplňující prostor, konkrétně **Hilbertovou křivkou**. V případě iterativní implementace Hilbertovi křivky se bude využívat binární reprezentace indexů bodů, kdy se v algoritmu v cyklu bude provádět bitový posun a budou se zjišťovat vždy poslední 2 bity, dle kterých se bude měnit pozice.

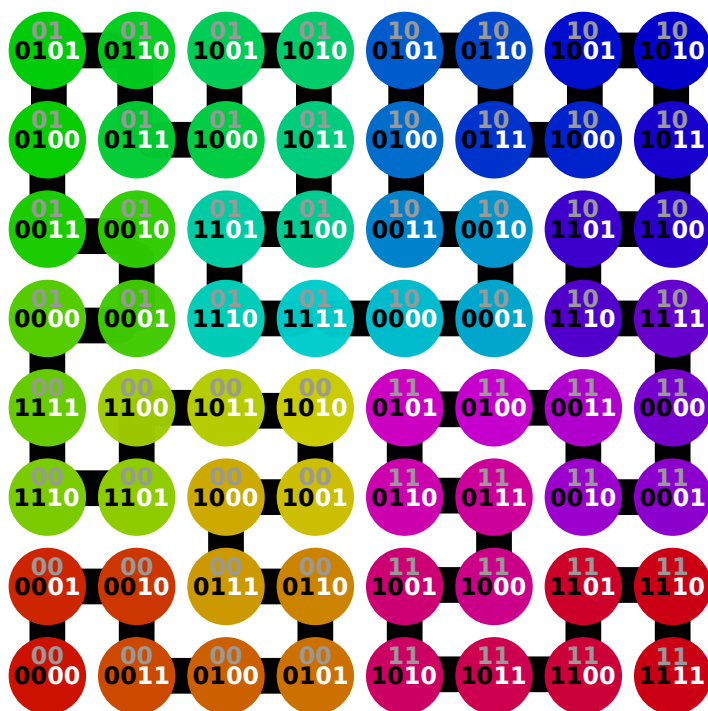
3 Popis řešení

Vykreslování probíhá ve třídě *Drawing*. Algoritmy fraktálů se nacházejí v balíčku *fractal*. Ke křivkám ze cvičení byla přidána Hilbertova křivka.

3.1 Hilbertova křivka

V práci je implementován algoritmus, který index bodu Hilbertovy křivky převádí do kartézské soustavy souřadnic bez použití rekurze. Algoritmus předpokládá, že bod s indexem 0 je na pozici $[0, 0]$. V první iteraci existují právě 4 body a počáteční křivka je pevně daná.

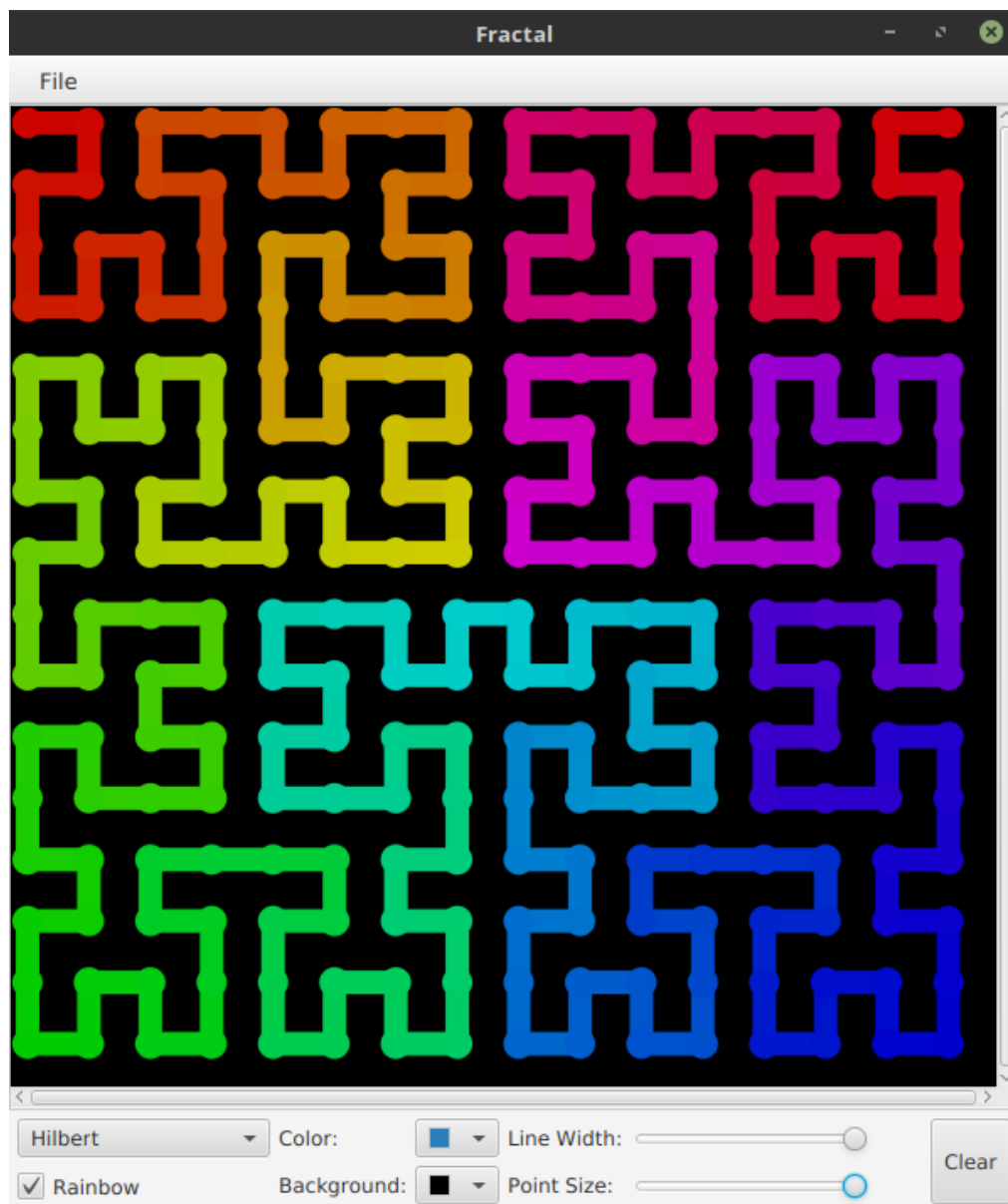
Pozice následujících bodů se dopočítávají podle indexu. Poslední 2 bity označují umístění bodu v rámci čtveřice bodů (tj. tvar křivky v první iteraci), následující dvojice bitů označuje umístění předešlé čtveřice bodů v rámci šestnáctice bodů (tj. křivka v druhé iteraci).



Obrázek 1: Indexy bodů v třetí iteraci

4 Uživatelská dokumentace

Spuštění aplikace se provede souborem `Fractal.jar`, který se nachází ve složce *App*. Po spuštění je nutné vybrat v dolním panelu křivku. Iterace se provede kliknutím na libovolné místo ve vykreslovací oblasti.



Obrázek 2: Okno aplikace

5 Závěr

U aplikace není vyřešeno omezení na určitý počet iterací fraktálu a proto u vyšší iterace může dojít ke zpomalení počítače či k vyčerpání dostupné paměti a pádu aplikace.

6 Reference

Iterative algorithm for drawing Hilbert curve – Marcin Chwedczuk. [online].

Dostupné z: marcin-chwedczuk.github.io/iterative-algorithm-for-drawing-hilbert-curve