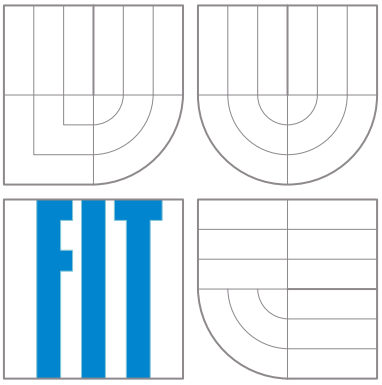


SFC - Soft Computing

Vizualizace Juliových množin

[129] C/C++

26. listopadu 2014

Autor: Bc. Radek Ševčík, xsevci44@stud.fit.vutbr.cz
Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

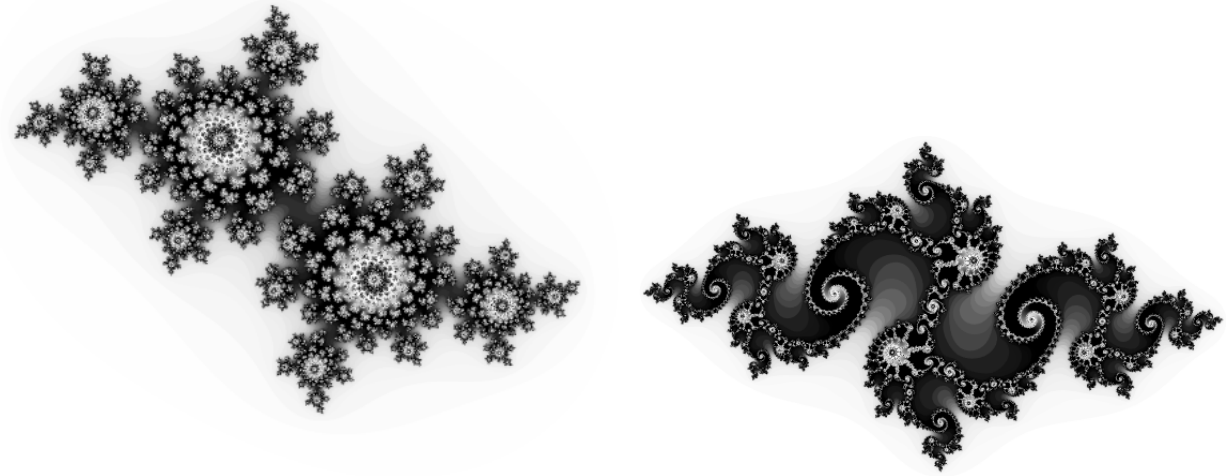
Obsah

1 Úvod

Cílem této práce je vytvořit program v jazyce C/C++, který umožní procházet Juliovy množiny.

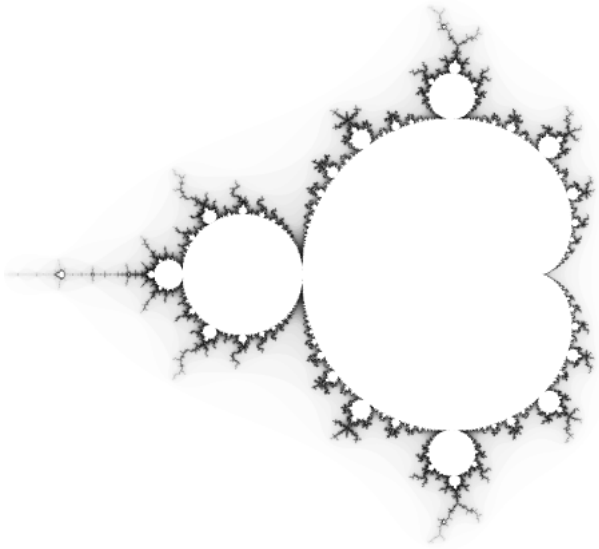
Juliova množina je množinou bodů z v komplexní rovině, kde její hranice tvoří fraktál. Tato množina je charakterizována parametrem, komplexním číslem c, a je vytvořena posloupností $z_{n+1} = z_n^2 + c$ takovou, pro které bod z nediverguje, tj. $|z| \leq 2$.

Juliova množina má těsný vztah k Mandelbrotově množině, která leží uvnitř kruhu se středem v 0 a poloměrem 2. Každému bodu Mandelbrotovy množiny odpovídá Juliova množina s parametrem daným souřadnicemi bodu komplexní roviny. Vizuálně nejzajímavější Juliovy množiny odpovídají bodům poblíž hranice Mandelbrotovy množiny.



(a) $c = -0,4 + 0,6i$ (b) $c = -0,775684 + 0,136367i$

Obrázek 1: Juliovy množiny



Obrázek 2: Mandelbrotova množina

2 Výpočet

Posloupnost $z_{n+1} = z_n^2 + c$ se může spočítat iterativně opakovaným umocňováním a přičítáním komplexního čísla. Výpočet se však musí po určité době zastavit a předpokládat, že daný bod komplexní roviny do množiny patří.

Čtverec komplexního čísla v algebraickém tvaru se spočítá jako $(a + bi)^2 = (a^2 - b^2) + (2ab)i$. Součet dvou komplexních čísel je triviální. Absolutní hodnota se spočítá jako $|z| = \sqrt{a^2 + b^2}$. K urychlení výpočtu dojde umocníme-li nerovnici, proto upravíme tvar $|z| \leq 2$ na skalární součin $z \cdot z \leq 4$.

Počet iterací ovlivňuje jak dobu výpočtu, tak přesnost zobrazení. Výpočet se prodlouží i tím, že je vhodné zajít více do detailů, tedy si zobrazení přiblížit. Je vhodné komplexní parametr c měnit plynule tak, aby byla zachována plynulost zobrazení, tj. snímková frekvence. To vyžaduje velký výpočetní výkon.

Jelikož vypočítáváme posloupnost pro každý obrazový bod, který následně zobrazíme v okně, čili provádíme rasterizaci obdelníku, můžeme přenechat tento úkol přímo na grafické kartě. Nemusíme se tedy starat o rasterizaci jako takovou. Dnešní grafické karty mají programovatelnou pipeline, tedy můžeme jednoduše zapsat výpočet posloupnosti a vykreslení jako shader. Pro výpočet barvy bodu se mohou použít různé algoritmy, v rámci projektu si vystačíme s jednoduchou barevnou paletou a indexem bude číslo, ve které iteraci bylo zjištěno, že bod diverguje.

Jako grafické API je použito *OpenGL 2.0* (2004) se shader jazykem *GLSL 1.10.59*. Pro zpřesnění výsledku se použijí 64bitová čísla v plovoucí řádové čárce namísto 32bitových tehdy, pokud to grafická karta podporuje, pomocí rozšíření *ARB_gpu_shader_fp64*. K vytvoření okna aplikace poslouží knihovna *glut* a ke správě *OpenGL* rozšíření pak knihovna *glew*.

Nejdříve je třeba nastavit viewport a projekční matici, pak nahrajeme barevné palety RGB jako 1D textury a nastavíme shadery. Zkompilujeme vykreslení čtverce na základní rozměry okna $[-1; -1], [1; 1]$ jako triangle strip do display listu, který pak budeme vykreslovat s daným shaderem.

Program ve vertex shaderu, který transformuje souřadnice $[0; 0], [1; 1] \mapsto [-2,5; -2,5], [2,5; 2,5]$ do proměnné *position* může vypadat například:

```
varying vec2 position;
void main(void) {
  position = (gl_MultiTexCoord0.st - 0.5) * 5.0;
  gl_Position = ftransform();
}
```

Při této transformaci je třeba brát v úvahu následné zpracování a výpočet obrazových bodů na souřadnice komplexní roviny při práci s myší.

Následně při rasterizaci použijeme fragment shader, který přiřadí každému bodu barvu z barevné palety a bere v potaz přiblížení a střed projekce:

```
uniform sampler1D tex;
uniform vec2 c;
uniform int iter;
uniform float zoom;
uniform vec2 center;
varying vec2 position;

vec2 f(in vec2 z) {
    return vec2(z.x * z.x - z.y * z.y, 2.0 * z.x * z.y) + c;
}

int julia(in vec2 z) {
    for (int i = 0; i < iter; ++i) {
        if (dot(z, z) > 4.0) return i;
        z = f(z);
    }
    return 0;
}

void main(void) {
    vec2 z = position * zoom + center;
    int n = julia(z);
    gl_FragColor = texture1D(tex, (float(n) / 100.0));
}
```

3 Kompilace a spuštění

Projekt je rozdělen do několika podadresářů:

- /bin – spustitelný program
- /doc – zdrojové kódy dokumentace
- /lib – knihovny třetích stran
- /lib/freeglut – OpenGL Utility Toolkit
  `http://freeglut.sourceforge.net/`
- /lib/glew-1.11.0 – OpenGL Extension Wrangler Library
  `http://glew.sourceforge.net/`
- /src – zdrojové kódy programu

Pro kompilaci pod operačním systémem *Debian GNU/Linux* je potřeba mít nainstalované balíčky `freeglut3-dev`, `libglew-dev` a samozřejmě `gcc` a `make`. Samotná kompilace pak proběhne zadáním příkazu `make` z adresáře `src`. Výstupem je spustitelný soubor `sfc2014`.

Pod operačním systémem *Microsoft Windows* probíhá kompilace z příkazové řádky spuštěním *Visual Studio 2013 → Visual Studio Tools → VS2013 x86 Native Tools Command Prompt*, přesunem do adresáře `src` a zadáním příkazu `NMAKE /f Makefile.msvc12`. Výstupem je spustitelný soubor `sfc2014.exe`, ke kterému je potřeba nakopírovat knihovní soubory `freeglut.dll` a `glew32.dll` z adresáře `lib`. Tímto způsobem byl vytvořen adresář `bin`.

Po spuštění programu se objeví okno s následujícím ovládáním:

- 0 – zobrazí nebo schová nápovědu
- 1 – vybere další předdefinovaný parametr c
- 2 – změní barevnou paletu
- 3 – přepne zobrazení mezi Mandelbrotovou a Juliovou množinou
- $+/-$ – změní počet iterací výpočtu
- kliknutí na levé tlačítko myši – nastaví příslušný bod jako střed
- pohyb kolečkem na myši – přiblížení nebo oddálení
- pohyb po obrazovce za stálého držení pravého tlačítka myši – změna parametru c podle směru pohybu

4 Závěr

Již v úvodu byli použity obrazy množin, které byly vytvořeny tímto programem se zapnutým vyhlazováním a následným post-processingem ve formě inverze barevné palety.

V rámci implementace jsou předdefinovány tři barevné palety a několik zajímavých komplexních parametrů c.

Zobrazení Mandelbrotovy množiny v programu byla snadnou úpravou fragment shaderu provedena ze zajímavosti a tato funkcionalita byla programu ponechána.