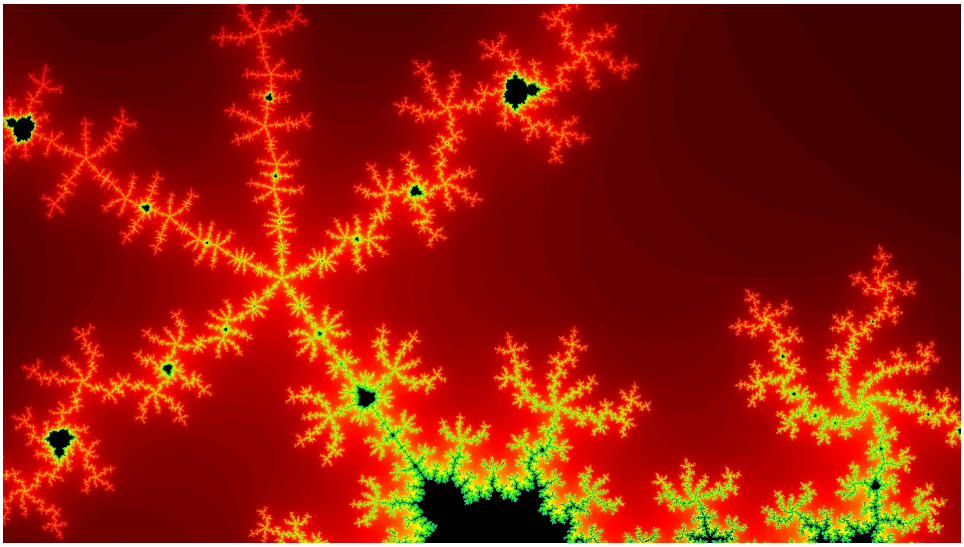
Mandelbrot

Oscar Melin

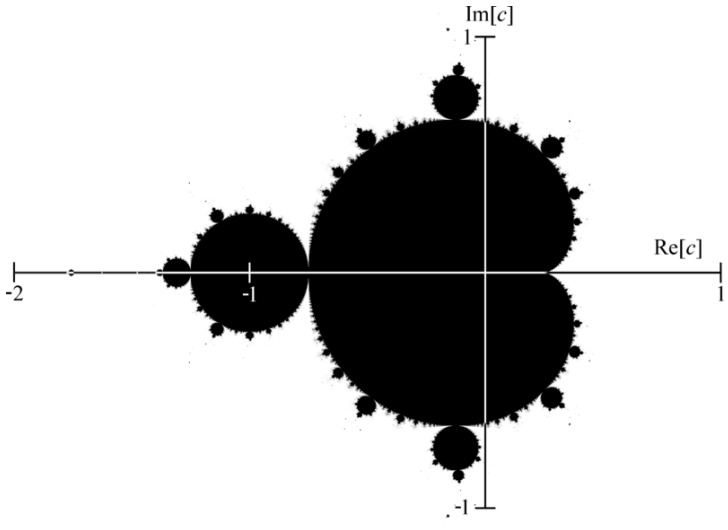
2016-02-15



Figur 1: Exempel på bild genererad med hjälp av mandelbrotmängden.

1 Uppgiften

Den här rapporten behandlar skapandet av en bildgenerator som genererar bilder baserade efter mandelbrotmängden (eng: Mandelbrot set). Figur 1 visar ett exempel på hur denna visualisering kan se ut. De svarta områdena motsvarar värden i mandelbrotmängden medans de färgglada områdena motsvarar värden utanför mandelbrotmängden. Kapitel 2 introducerar teorin bakom mandelbrot samt hur jag implementerat min mandelbrot-bildgenerator i del 2.1 respektive 2.2. Kapitel 3 sammanfattar med utvärdering och reflektioner.



Figur 2: Bild som visar hela mandelbrotmängden, Re[c] och Im[c] betecknar den reella respektive imaginara delen av c. Källa: Wikipedia.

2 Ansats

2.1 Teori

Mandelbrotmängden är definierad som mängden komplexa tal för vilka funktionen $f(z) = z^2 + c$ inte går mot oändligheten (divergerar) när den itereras med startvärdet $z = 0$ där c är ett komplext tal, $c = a + bi$. Exempel:

$$c = 0$$

$$f_0(0) = 0^2 + 0$$
$$f_1(0) = 0^2 + 0$$

$$\dots$$

$c = 0$ är alltså en del av mandelbrot mängden medans till exempel $c = 1$ ger den divergerande serien:

$$f_0(0) = 0^2 + 1$$
$$f_1(1) = 1^2 + 1$$
$$f_2(2) = 2^2 + 1$$

$$\dots$$

$c = 1$ är alltså utanför mandelbrotmängden. Figur 2 ger en bild av vart $c = 0$, $c = 1$ ligger i relation till hela mängden.

Alltså, för att veta om ett komplext tal z ligger i mandelbrotmängden skulle vi behöva iterera **väldigt** många gånger för att se om $f(z)$ närmar

sig oändligheten. Lyckligtvis för oss är mandelbrotmängden kompakt dvs. att den är både sluten och begränsad. Denna gräns är vid $|f_n(z)| < 2$ vilket innebär att om $|f_n(z)| >= 2$ så vet vi att f divergerar och c inte är en del av mandelbrotmängden. Praktiskt sett innebär detta att vi kan sätta ett tak för hur många gånger vi itererar över f och om vi går över 2 så är c definitivt inte med och om vi stannar under så är c möjligen med i mandelbrotmängden. Vilken färg vi väljer bestäms av hur många gånger vi måste iterera innan $|f_n(z)| >= 2$, vilket betyder att det är hur snabbt värden som ligger utanför mandelbrotmängden växer som ger oss de vackra färgerna och inte värdena innuti.

2.2 Praktik

Implementeringen är uppdelad i fem moduler: Cmplx 2.2.1, Brot 2.2.2, ppm 2.2.3, Color 2.2.4 och Mandel 2.2.5.

2.2.1 Cmplx

Första modulen cmplx hanterar grundläggande metoder för att hantera complexa tal. Den exporterar funktioner för att skapa, addera, kvadrera och ta absolutvärdet av ett komplext tal $\{X, Y\}$ där X är den reella delen och Y den imaginära.

2.2.2 Brot

Modulen Brot exporterar en funktion *mandelbrot/2* som givet ett komplext tal C och max antal iterationer M returnerar antaler iterationer I som det krävdes för att $|f| >= 2$ alternativt 0 ifall iterationsgränsen nås.

```
mandelbrot(C, M) ->
  Z0 = cmplx:new(0, 0),
  I = 0,
  test(I, Z0, C, M).

test(_, _, _, 0) ->
  0;
test(I, Z0, C, M) ->
  Z = cmplx:add(cmplx:sqr(Z0), C), % Zn+1 = Zn^2 + c
  Abs = cmplx:abs(Z),

  if
    Abs >= 2 -> I;
    true -> test(I+1, Z, C, M-1)
  end.
```

2.2.3 ppm

ppm exporterar funktionen write(Name, Image) som tar en matris av rgb-värden och genererar en *.ppm* bild.

2.2.4 Color

För att bestämma vilken färg varje punkt tilldelas utgår vi från hur många iterationer som krävdes för att fastställa ifall punkten tillhör mängden. Funktionen convert/2 tar Depth (antal iterationer), Max (max antal iterationer) för en punkt och räknar ut motsvarande färg.

```
-module(color).
-export([convert/2]).

convert(Depth, Max) ->

A = (Depth/Max)*4,
X = trunc(A),
Y = trunc(255*(A - X)),

case X of
  0 -> {Y, 0, 0};
  1 -> {255, Y, 0};
  2 -> {255 - Y, 255, 0};
  3 -> {0, 255, Y};
  4 -> {0, 255 - Y, 255}
end.
```

2.2.5 Mandel

Modulen mandel är själva hjärtat i den här hårddisken och ansvarar för att genera vårt kompletta mandebrotset. Funktionen mandelbrot/6 tar argumenten Width, Height, X, Y, K, Depth och returnerar en lista av listor med tupler där varje tuple-element representerar ett RGB-värde. Width och Height är dimensionen på bilden vi vill generera, X och Y är positionen för bildens övre vänstra hörn, K är offset och Depth är max antal iterationer.

3 Utvärdering och Sammanfattning

Sammanfattningsvis har den här uppgiften varit väldigt engagerande och intressant. Många timmar har gått åt att leka med färgschemat samt ändra koordinater i jakt på coola mönster. Programmeringsmässigt så tycker jag inte att det var mycket nytt utan det mest intressanta var själva Mandelbrotmängden.

I framtiden vore det intressant att ta en titt på besläktade mängder, till exempel Juliamängden.