

Julia set in C# (Windows form)  
Dedicated to Julia Slipchuk

Author: O. Hryshchuk

16 липня 2019 р.

# 1 Алгоритм побудови

## 1.1 Опис

Вікіпедія про дану множину говорить наступне:

*У голоморфній динаміці, множина Жуліа  $J(f)$  раціонального відображення  $f : \mathbb{CP}^1 \rightarrow \mathbb{CP}^1$  — множина точок, динаміка в околиці яких у певному сенсі нестійка відносно малих збурень початкового положення.*

Занурення в теорію функцій комплексних змінних справа довга й невесела тому обмежусь наступним: Суть алгоритму полягає у переборі всіх точок на екрані (в даному випадку було використано `Bitmap`). Кожна точка екрану задає своїми координатами комплексне число  $z = z_x + iz_y$  і відповідно функцію  $f(z) = z^2 + c$ . Для побудови фракталу необхідно перевірити дві умови: чи входить точка з координатами  $(z_x, z_y)$  в коло з радіусом 2, і друга умова - якщо була виконана максимально можлива кількість ітерацій, для того щоб отримати такі  $z_x$  та  $z_y$ , які задовольняють першу умову, то зафарбовуємо цю точку.

Числа  $z_x$  та  $z_y$  на початку отримуються масштабування координат точки на екрані  $(x, y)$ , а далі змінюються за наступними правилами:

$$\begin{aligned} tmp &= z_x^2 - z_y^2; \\ z_y &= 2z_x z_y + c_y; \\ z_x &= tmp + c_x; \end{aligned} \tag{1}$$

Де  $c = c_x + c_y i$

## 1.2 Псевдокод

```
for each pixel (x, y) on the screen, do:
{
    zx := scaled x coordinate of pixel

    zy := scaled y coordinate of pixel

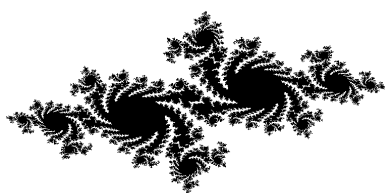
    iteration := 0
    max_iteration = set max iteration

    while (zx * zx + zy * zy < 4 AND iteration < max_iteration)
    {
        xtemp := zx * zx - zy * zy
        zy := 2 * zx * zy + cy
        zx := xtemp + cx

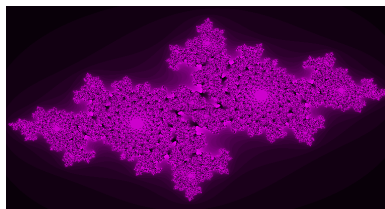
        iteration := iteration + 1
    }

    if (iteration = max_iteration)
        draw black pixel
}
```

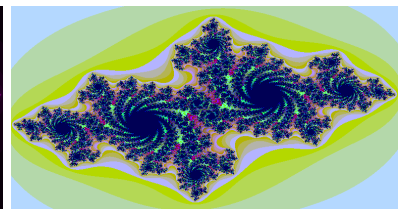
На виході такого алгоритму отримуємо чорнобіле зображення, наприклад як те, що на рис. 1.а.



(a) Black



(б) Purple



(в) Color

Рис. 1: Julia set for  $c = -0.7 + 0.27i$

Для того щоб розфарбувати точки, потрібно враховувати скільки ітерацій було виконано до завершення циклу, і відповідно відмалбовувати всі точки. Спосібів як це робити є багато. Той що зображений на рис 1.б у `C#` має такий вигляд:

```
bitmap.SetPixel(x, y, Color.FromArgb(255, (i * 9) % 255, 0, (i * 9) % 255));
```

Другий варіант більш досконалий в плані кольору, але має обмеження на кількість ітерацій. Він генерує масив кольорів з 256 елементів, а потім ці елементи використовуються для замальовування:

```
Color[] colors = (from c in Enumerable.Range(0, 256)
                  select Color.FromArgb((c >> 5) * 36, (c >> 3 & 7) * 36, (c & 3) * 85)).ToArray();
```

### 1.3 Etc

Раджу звернути увагу на параметри, які задають число  $c$ , змінивши їх можна отримати абсолютно різні фігури. Також можна погратися з масштабуванням координат, це може вирізати доволі цікаві частини фракталу. На рис. 1 масштабування задавалося наступним чином:

```
zx = 1.5 * (x - w / 2) / (0.5 * zoom * w);
zy = 1.0 * (y - h / 2) / (0.5 * zoom * h);
```

Де  $w$  та  $h$  це ширина та висота поля, на якому відбувається малювання,  $zoom$  позначає масштаб рисунка, по замовчуванню рівний 1.

### 1.4 End

Що ж, на цьому поки що все, залишайю простір тобі для фантазій. У випадку питань - звертайся. Бажаю успіху з написанням коду!