Université de Strasbourg
Algorithmique et Programmation 1 — Feuille de TP

Fractales

Les formes fractales.

Ce sont des formes intrinsèquement récursives. Le triangle de Sierpinski est une fractale (figure 1) qui vérifie la règle récursive suivante : le triangle de Sierpinski (a, b, c) est l'union des trois triangles de Sierpinski (a, m_b, m_c), (m_a, b, m_c) et (m_a, m_b, c), où m_a, m_b et m_c sont les milieux de $[bc]$, $[ac]$ et $[ab]$ respectivement.

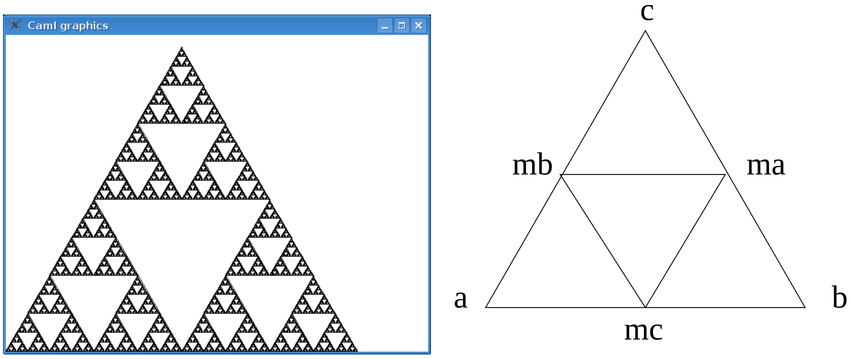

FIG. 1 – Triangle de Sierpinski.

La géométrie.

La géométrie en 2D est basée sur des objets de type `point`, défini avec son constructeur par :

```
type point = {x:int; y:int};;
let (creer_point : int -> int -> point) =
function x1 -> function y1 -> {x=x1; y=y1};;
```

Exercice 1 Triangle de Sierpinski.

1.1 étude du code fourni. Le fichier `sierpinski-enreg.ml` contient le code utile au tracé du triangle de Sierpinski, excepté la fonction récursive elle-même (`let rec sierpinski a b c`). Étudiez ce code pour comprendre à quoi sert chaque fonction.

Indication : on pourra interpréter tout le code d'un coup à l'aide de la commande
`#use "sierpinski-enreg.ml";;`

1.2 implémentation. La définition mathématique du triangle implique une récurrence infinie. À l'inverse, l'algorithme récurrent a besoin d'une condition d'arrêt : quand un triangle de Sierpinski ne peut plus être distingué visuellement (sa taille est inférieure à 2 pixels), on le dessinera comme un triangle simple. Il en résulte l'algorithme 1. Programmez cet algorithme et testez le programme.

Indication : pour exécuter plusieurs instructions à la suite on peut utiliser la syntaxe suivante

```
begin
  instruction_1 ;
  instruction_2 ;
  ...           ;
  instruction_n
end
```

1.3 la taille limite. Testez le programme avec les conditions d'arrêt suivantes : 14 puis 15, 16 et 17 pixels. Expliquez le résultat obtenu.

1.4 version flottante. Jusque là les coordonnées des points sont des entiers. Modifiez le code pour travailler sur des coordonnées flottantes. Pour cela, il faut corriger le type `point`, son constructeur, les fonctions `milieu`, `taille` et `sierpinski`.

ALGO. **1** Algorithme de tracé du triangle de Sierpinski (a, b, c).
On appelle m_a, m_b et m_c les milieux de $[bc]$, $[ac]$ et $[ab]$.

si le triangle (a, b, c) est trop petit pour être distingué **alors**
  Dessiner le triangle (a, b, c) simple.
sinon
  Tracer les triangles de Sierpinski (a, m_b, m_c), (m_a, b, m_c), et (m_a, m_b, c).
fin si

Exercice 2 Carrécursifs

On souhaite dessiner des fractales à base de carrés en utilisant des fonctions récursives. Pour cela :

2.1 Le fichier `carrecursifs.ml` contient les définitions utiles.
Étudiez ce code pour comprendre à quoi sert chaque fonction.

2.2 Déterminez les algorithmes permettant de dessiner les carrés représentés par les figures 2 et 3.

2.3 Implémentez les en Caml.

2.4 Déduisez un algorithme qui permet de dessiner le carré représenté par la figure 4.

2.5 Implémentez le.

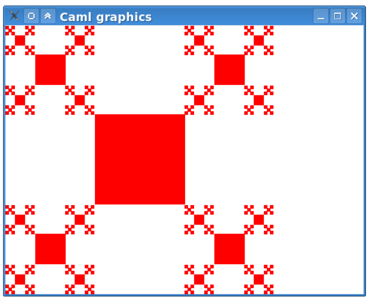

FIG. 2 – Carré récursif 1

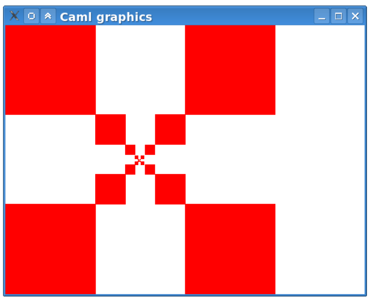

FIG. 3 – Carré récursif 2

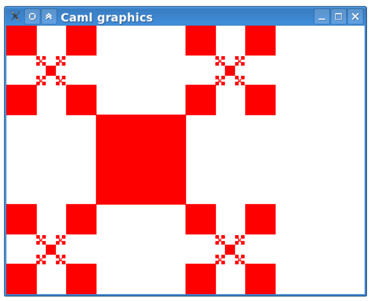

FIG. 4 – Carré récursif 3