▶ **HX4 & HX1 – Option Informatique**
Quatrième TP Caml
Vendredi 11 février 2011

Mandelbrot

1 Préliminaires

1.1 Complexes

▶ **Question 1** *Écrire les fonctions classiques sur les complexes en Caml : addition, multiplication, conjugué, module carré. Une bonne mqnière est de considérer les complexes comme des float*float.*

1.2 Graphiques

Dans cette partie, on utilise les fonctions suivantes :

```
set_color : color -> unit
rgb: int -> int -> int -> color
plot : int -> int -> unit
```

qui servent respectivement à, définir la couleur de dessin, fabriquer une couleur en fonction de ses composantes rouge, verte et bleue, colorier un point.

▶ **Question 2** *Dessiner un carré de taille 255x255 dont dont la couleur varie graduellement entre bleu (coin inférieur gauche, 0 0 255), magenta (inférieur droit, 255 0 255), vert (supérieur gauche 0 255 0) et jaune (supérieur droit 255 255 0).*

2 Petits exercices techniques

Si vous vous sentez à l'aise avec caml, passez cette partie.

▶ **Question 3** *Écrire une fonction entremele qui prend deux listes* $[a_1; a_2...]$ *et* $[b_1; b_2; ...]$ *et qui renvoie* $[(a_1, b_1); (a_2, b_2); ...]$. *On peut supposer que les deux listes ont la même taille.*

▶ **Question 4** *Écrire une fonction*

```
flatten : a' list list -> a' list
```

qui prend une liste de listes (de la forme $[[a_1; a_2; ...; a_n]; [b_1; b_2; ...]; ...]$ *et la transforme en une liste de la forme* $[a_1; a_2; ...; a_n; b_1; b_2; ...]$.

3 Fractales de Mandelbrot

On définit l'ensemble de Mandelbrot comme : $M = \{c \in \mathbb{C} : \exists s \in \mathbb{R}, \forall n \in \mathbb{N} | P_c^n(0)| < s\}$ avec $P_c(z) = z^2 + c$. En fait on peut se restreindre à $s = 2$ et on va utiliser cette propriété pour dessiner la fractale et la colorer en fonction du plus petit n tel que $|P_c^n(z)| \geq 2$.

▶ **Question 5** *Écrire une fonction qui prend deux arguments m et c et qui renvoie* $min(m, inf_{n\in\mathbb{N}} n, |P_c^n(0)| \geq 2)$.
On a maintenant tous les outils pour dessiner une belle fractale de Mandelbrot : on choisit un N assez grand (mais pas trop car plus N est grand, plus le temps de calcul est long) et on dessine un carré - comme dans l'exercice 2 - mais cette fois, la couleur du pixel x,y dépend du nombre d'itérations pour que $P_{c(x,y)}(0)$ soit plus grand que 2 en module. c est choisie de façon à voir $[-2, 1]$ sur l'axe des x et $[-1, 1]$ sur celui des y.

▶ **Question 6** *Dessiner la fractale de Mandelbrot. Faire varier les manières de colorier. Zoomer sur certains points.*

4 Entiers

Pour programmer il est pratique d'avoir pleins d'objets de types différents : entiers, flottants, fonctions, etc ... Cependant quand on veut raisonner sur la nature des langages de programmation (ce qu'ils sont capables de calculer) il est plus simple d'avoir le moins d'objets possible. Les entiers de Church sont une définition des entiers à l'aide des fonctions. On définit pour $n \in \mathbb{N}$ l'entier n' qui a f associe f^n.

▶ **Question 7** *Écrire deux fonctions church_int et int_church qui effectuent les correspondances entre entiers de caml et entiers de church.*

▶ **Question 8 *** *Écrire les fonctions : addition, multiplication et exponentiation sur les entiers de church (sans repasser aux entiers bien évidemment)*

▶ **Question 9 *** *Écrire une fonction qui prend en argument une chaine entièrement parenthésée (de la forme* $((4*6)+(6*3))$*) et qui calcule ce qu'elle vaut. On pourra faire une fonction qui lit un entier et renvoie à la fois la valeur de l'entier et jusqu'où il a lu dans la chaine, et une seconde qui lit une expression et qui renvoie la valeur de celle ci ainsi que jusqu'où elle a lu dans la chaine.*
■

▶ **HX4 & HX1 – Option Informatique**
Quatrième TP Caml
Vendredi 11 février 2011

Mandelbrot

Un corrigé

▶ Question 1

```
let add (x,y) (a,b) = (a+.x,y+.b)
let mul (x,y) (a,b) = (x*.a-.y*.b,a*.y+.b*.x)
let conj (x,y) = (x,-.y)
let modu a = fst (mul a (conj a))
```

▶ Question 2

```
let carre =
  clear_graph();
  for x = 0 to 255 do
    for y = 0 to 255 do
      set_color (rgb x y (255 - y));
      plot x y;
    done;
  done;;
```

▶ Question 3

```
let rec entremele l l' = match (l,l') with
  | ([],a) -> []
  | (a,[]) -> []
  | (a::q,x::t) -> (a,x)::entremele q t
```

▶ Question 4

```
let flatten l =
  let rec foo res = function
  | [] -> res
  | a::q -> foo (a@res) q
  in
foo [] l

(* ou en detaillant le @ : *)

let flatten l =
  let rec ajoute l = function
    | [] -> l
    | a::q -> ajoute (a::l) q
  in
  let rec foo res = function
    | [] -> res
    | a::q -> foo (ajoute res a) q
  in
  foo [] l
```

▶ Question 6

```
let diverge m c =
  let rec foo v = function
    | 0 -> 0
    | i -> if modu v >= 2.0 then i else foo (add (mul v v) c) (i-1)
  in
  m-(foo (0.,0.) m)

let dessine =
  let eps = 0.03 in
  let x = -1.75 in
  let (ymax,ymin,xmax,xmin) = (0.+.eps,0.-.eps,x+.eps,x-.eps) in
  set_color black ;
  clear_graph () ;
  for x = 0 to 600 do
    for y = 0 to 400 do
      let n = diverge 42 (((float_of_int x)/.600.)*.(xmax-.xmin)+.xmin,
      ((float_of_int y)/.400.)*.(ymax-.ymin)+.ymin) in
      if n < 42
      then (set_color (rgb (n*2) (n*3) n) ; plot x y)
    done
  done
```

▶ Question 8

```
let church_int n f x =
  let rec foo = function
    | 0 -> x
    | i -> f (foo (i-1))
  in
  foo n

let int_church n = n (fun x -> x+1) 0

let add n m f x = n f (m f x)
let mul n m f x = n (m f) x
let exp n m = m n
```

▶ Question 9

```
let eval s =
  let rec lit_entier v i =
    if s.[i] >= '0' && s.[i] <= '9'
    then lit_entier (v*10+(int_of_char s.[i]-int_of_char'0')) (i+1)
    else (i,v)
  in
  let rec lit_expr i =
    if s.[i] <> '('
    then lit_entier 0 i
    else
        let (s1,v1) = lit_expr (i+1) in
        let (s2,v2) = lit_expr (s1+1) in
        (s2+1,match s.[s1] with
          | '+' -> v1+v2
          | '-' -> v1-v2
          | '*' -> v1*v2
          | '/' -> v1/v2)

      in
    snd (lit_expr 0)
```