

CONTROLE FLASH Numéro 2 (PF1)**50 minutes****DATE : mardi, à 8h****SECTION : Groupe 4, L1-Info****NOM :****PRENOM :**

Question 1a : On appelle **polygone** une liste de points du plan, chaque point étant représenté par une structure de type `posn` définie par : `(define-struct posn (x y))`. Programmez de manière ***réursive enveloppée*** une fonction `(polygone n)` retournant un polygone de longueur `n` contenant des points choisis au hasard dans un canvas 300 x 300. Exemple :

`(polygone 3) → (#(struct:posn 16 29) #(struct:posn 188 133) #(struct:posn 36 250))`

`(define (polygone n)` ; récurrence enveloppée

Question 1b : Programmez la fonction `(polygone n)` de manière *itérative*.

`(define (polygone n)` ; itération
`(local`

Question 1c : Programmez la fonction `(polygone n)` *en une ligne*, sans récurrence ni itération.

`(define (polygone n)`

Question 2a : La fonction ci-dessous *n'est pas itérative*. A quoi le voyez-vous précisément ?

```
(define (iterer f n x)
  (if (= n 0) x (f (iterer f (- n 1) x))))
```

Question 2b : Que vaut (iterer sqr 3 2) ?

Question 2c : **Sans** utiliser le mot `local`, donnez une version *itérative* de la fonction `iterer`.

Question 3 : **SYSTEME DE PARTICULES**. Le *fond* de l'animation sera un canvas jaune 400 x 400 dans laquelle on placera les segments AB et CD en noir (cf figure).

```
(define FOND
```

Une **balle** sera représentée par une structure `(x,y,dx,dy,col)` où `x,y` est la position, où `dx,dy` est la vitesse et `col` la couleur : `(define-struct balle (x y dx dy col))`. Définissez une fonction `(random-balle)` retournant une balle située en A avec une vitesse initiale *verticale* choisie au hasard dans `[2,10]`. La couleur sera choisie au hasard parmi `"red"` et `"blue"`.

```
(define (random-balle)
```

Le **monde initial** est une liste `L` de 30 balles obtenues par `random-balle`.

```
(define INIT
```

Programmez la fonction `dessiner` qui construit la scène correspondant au monde `L` à un moment donné. Une balle aura un rayon de 10.

(define (dessiner L) ; monde → scène

Il n'y a *aucune gravitation* dans cette animation. Les balles vont descendre à vitesse constante du point A jusqu'à B. Au point B, les balles bleues vont à gauche et les balles rouges à droite, sans changer de vitesse, seulement de direction. Lorsqu'une balle *rouge* atteint le bord droit, elle meurt et est supprimée du monde. Lorsqu'une balle *bleue* atteint le bord gauche, elle renaît en A avec *random-balle* donnant l'illusion d'un flot continu mais diminuant. Programmez la fonction d'évolution suivant en respectant dans un `cond` et dans l'ordre les étapes ci-dessous :

- a) elle va sortir du bord gauche en C
- b) elle va sortir du bord droit en D
- c) elle avance à l'horizontale et n'est plus sur AB
- d) elle descend et va dépasser B !
- e) sinon (elle descend entre A et B)

(define (suivant L) ; monde → monde

Programmez enfin la fonction qui **termine** l'animation lorsqu'il n'y a plus de balles !

```
(define (final? L) ; monde → boolean
```

Le *big-bang* vous est gracieusement offert par la maison.

```
(big-bang INIT  
  (on-tick suivant)  
  (on-draw dessiner)  
  (stop-when final?))
```

