

SAÉ S1.01 : implémentation d'un besoin client

Le jeu du Snake : version 1

Votre mission

Dans cette première version, vous devez programmer le déplacement d'un serpent de taille $N^{(1)}$: le serpent devra se déplacer vers la droite tant que l'utilisateur n'a pas appuyé sur la touche 'a'.

À l'exécution, l'utilisateur choisira d'abord la position initiale du serpent⁽²⁾ en tapant un numéro de colonne x, puis un numéro de ligne y. L'écran devra ensuite s'effacer et afficher le serpent en mouvement.

⁽¹⁾ Vous prendrez $N=10$.

⁽²⁾ En fait, l'utilisateur fournira la position initiale de la tête du serpent, les autres anneaux du serpent seront ajoutés à sa gauche. Ces "coordonnées" devront être comprises entre 1 et 40.

Vous déposerez sur Moodle, avant la date convenue, votre code source "version1.c" et le cahier de tests correspondant complété.

Note : votre code source devra respecter les **conventions de codage** vues en R1.01 (voir document sur Moodle) : commentaires, indentation, utilisation de constantes, nommage des variables, etc.

Consignes

Votre programme devra obligatoirement contenir les procédures/fonctions décrites ci-dessous.

- `void afficher(int x, int y, char c)`

Cette procédure doit afficher le caractère `c` à la position `(x, y)`, où `x` représente le numéro de colonne et `y` le numéro de ligne.

- `void effacer(int x, int y)`

Cette procédure doit afficher un espace à la position `(x, y)`, où `x` représente le numéro de colonne et `y` le numéro de ligne.

- `void dessinerSerpent(int lesX[], int lesY[])`

ou

- `void dessinerSerpent(int laPosition[][])`

Cette procédure sera chargée d'afficher à l'écran un à un les éléments du serpent dont les coordonnées sont fournies dans le(s) tableau(x) en paramètre(s).

- `void progresser(int lesX[], int lesY[])`

ou

- `void progresser(int laPosition[][])`

Cette procédure sera chargée de calculer et d'afficher la prochaine position du serpent. Par exemple, si le serpent a sa tête en (12, 3), la procédure doit faire en sorte que le serpent évolue d'une position vers la droite et donc l'affiche avec la tête en position (13, 3).

Détails techniques

- Le serpent est constitué d'une tête (représentée par un O) et d'anneaux (représentés par des X).

Exemple :

```
XXXXXXXXXO
```

Dans cet exemple, le serpent a une longueur de 10 : 9 anneaux + la tête. Les anneaux sont positionnés à gauche de la tête.

- La vitesse de déplacement dépendra d'une valeur de temporisation entre deux affichages successifs du serpent (voir fonction `usleep()` plus loin). Faites en sorte que le mouvement ne soit pas trop rapide.
- La structure de données nécessaire à la représentation du serpent est une suite de couples de coordonnées (x,y). Par exemple, cette position du serpent de taille 5 :

```
123456789...
1  XXXXO
2
3
4
```

est codée par la suite de coordonnées (7,2), (6,2), (5,2), (4,2), (3,2). On commence par les coordonnées de la tête, puis par le premier anneau, etc.

Cette suite de coordonnées peut être gérée :

➔ soit par deux tableaux à une dimension, l'un pour les 'x', l'autre pour les 'y'.

lesX :

7	6	5	4	3
---	---	---	---	---

lesY :

2	2	2	2	2
---	---	---	---	---

➔ soit par un tableau à deux dimensions :

laPosition 0

7	6	5	4	3
---	---	---	---	---

 1

2	2	2	2	2
---	---	---	---	---

Note : vous choisirez l'une de ces deux possibilités.

Compléments techniques en langage C

- Pour effacer l'écran, utilisez l'instruction
`system("clear");`
- Pour la temporisation, utilisez la fonction
`int usleep(int duree);`
Cette fonction interrompt l'exécution du programme pendant le nombre de microsecondes fourni en paramètre.
Pensez à inclure `<unistd.h>` au début de votre programme.
- Pour positionner le curseur à un endroit précis (avant l'affichage d'un caractère par exemple), utilisez la procédure
`void gotoXY(int x, int y);`
où x et y représentent respectivement le n° de colonne et le n° de ligne. Le code source de cette procédure est fourni sur Moodle.

Note : la numérotation des lignes et colonnes à l'écran démarre à 1.

- Pour vérifier s'il y a eu frappe d'un caractère au clavier, mais sans que cela bloque l'exécution, utilisez la fonction :
`int kbhit();`
dont le code source est fourni sur Moodle. Cette fonction teste s'il y a eu frappe d'un caractère au clavier et retourne 1 si c'est le cas et 0 sinon.
Pensez à inclure `<fcntl.h>`, `<unistd.h>` et `<termios.h>` au début de votre programme.

- Les tableaux à une dimension en langage C

Un tableau est une collection d'éléments de même type. La taille d'un tableau (son nombre de "cases") est fixé dès la déclaration et ne peut pas être modifié par la suite. Les éléments d'un tableau sont indicés de 0 à N-1, N étant la taille du tableau ;

Exemple : tableau de taille 10

0	1	2	3	4	5	6	7	8	9

Déclarations

- déclaration d'une variable de type tableau de 10 entiers :
`int tab[10];`
- déclaration d'un tableau de 50 réels :
`float tab[50];`

Note : une bonne pratique consiste à déclarer la taille d'un tableau en constante, sous la forme :

- `#define TAILLE 10`
- la déclaration du tableau devient donc :
`int tab[TAILLE];`

Accès aux éléments d'un tableau

- Chaque élément du tableau est accessible via son indice, situé entre des crochets :

```
int element;
element = tab[5]; // la variable 'element' contiendra
                  // la valeur du 6e élément du
                  // du tableau (le contenu de la case
                  // d'indice 5)
```
- Parcours complet d'un tableau : boucle FOR

```
#define TAILLE 10
int tab[TAILLE];
int indice;
for (indice=0 ; indice<TAILLE ; i++){
    // accès ou modif de tab[i]
}
```