

7 Gestion de l'écran

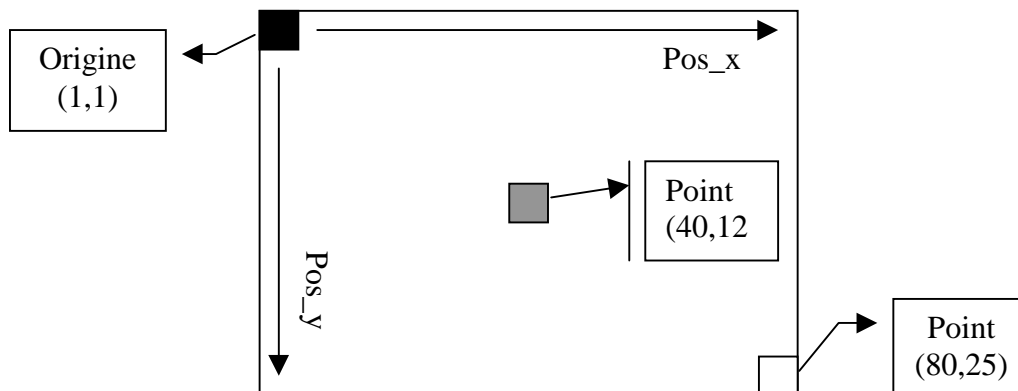
7.1 Prologue

Nous allons voir dans ce chapitre quelques fonctions permettant de se positionner et d'afficher à l'écran. Pour utiliser ces fonctions nous aurons besoin de la bibliothèque conio.h. (#include <conio.h>)

7.2 L'écran

7.2.1 Description

Un écran Dos se décompose de 80 colonnes et 25 lignes. Pour se positionner à un endroit précis de l'écran, on utilise la fonction **gotoxy (pos_x, pos_y)**;
pos_x allant de 1 à 80 et pos_y allant de 1 à 25.
L'origine est placée dans le coin haut gauche.



Pour faire afficher un caractère ou une chaîne de caractère à cet endroit, on utilise printf.

Exemple :

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

int main ()
{
    clrscr ();
    gotoxy (70,25);
    printf ("Coucou");
    getch ();
    return (0);
}
```

7.2.2 Exercice d'application n°1

Faire afficher en (12,7) "Bonjour c'est moi et j'utilise le C" puis afficher en (23,18) "C'est bien on peut faire ce que l'on veut".

7.3 Choix multiple : switch ... case

7.3.1 Définition

Switch ... case permet l'exécution d'une série d'instructions dans le cas où une variable a une valeur précise.

On peut traduire **switch ... case** par dans le cas où la variable vaut ... faire ...

En plus des cas où, il est possible d'ajouter le cas **default** qui signifie par défaut faire ...

Exemple commenté

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

int main ()
{
    char car;

    clrscr ();
    gotoxy (1,1);
    printf ("Saisie d'une touche : ");
    car = getch ();
    gotoxy (1,3);

    /* Réaliser les tests de cas sur la variable car */
    switch (car)
    {
        /* Dans le cas où car est égal au caractère a,e,i,o,u ou
        y afficher "Voyelle" */
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
        case 'y':
            printf ("Voyelle\n");
            /* Fin des instructions concernant un caractère a,e,i,o,u
            ou y */
            break;

        /* Dans le cas où car est égal au caractère 1 afficher
        "Chiffre 1" */
        case '1':
            printf ("Chiffre 1\n");
            break;

        /* Dans les autres cas afficher "Rien à faire pour ce
        cas" */
        default:
            printf ("Rien à faire pour ce cas\n");
            break;
    }

    getch ();
    return (0);
}
```

7.3.2 Exercice d'application n°2

Réaliser un programme qui fait afficher :

- ☞ "Gauche" lorsque l'on appuie sur g ou G
- ☞ "Droite" lorsque l'on appuie sur d ou D
- ☞ "Haut" lorsque l'on appuie sur h ou H
- ☞ "Bas" lorsque l'on appuie sur b ou B

7.4 Opérateurs unaires

Les opérateurs unaires (unary) du C sont des raccourcis mathématiques permettant de réaliser des opérations mathématiques ou logiques simples utilisant 2 fois la même variable. Ils facilitent aussi les calculs de la machine.

Exemple :

`a = a + 5;` s'écrit `a += 5;`

Quelques opérateurs

Opérateur	Signification	Exemple
<i>a += val</i>	$a = a + val$	<code>a += 3</code>
<i>a -= val</i>	$a = a - val$	<code>a -= 7</code>
<i>a *= val</i>	$a = a * val$	<code>a *= 9</code>
<i>a /= val</i>	$a = a / val$	<code>a /= 11</code>

Il existe bien d'autres opérateurs unaires notamment logique mais ceci dépassant l'étendue de ce cours, je vous renvoie aux livres qui traitent cela très bien.

7.5 Exercice : animation d'un curseur à l'écran

Traduire en langage C, compilez, exécutez, comprenez ...

Déclarer une variable x (initialisée à 10) de type entier.

Déclarer une variable y (initialisée à 10) de type entier.

Ces variables représenteront les positions du curseur à l'écran.

Déclarer une variable car de type caractère.

Déclarer une variable sortie de type entier (vrai ou faux en fait) initialisée à faux (0).

Faire

Effacer l'écran.

Afficher le caractère # en position x,y

Saisir un caractère sans écho et le mettre dans car.

Dans le cas où car est égal à :

G ou g : $x = x - 1$ (x--)

D ou d : $x = x + 1$ (x++)

H ou h : $y = y - 1$ (y--)

B ou b : $y = y + 1$ (y++)

S ou s : sortie = vrai

Autre : réaliser un beep

Fin du Dans le cas où

Tant que (sortie = faux) /* ou !sortie */

Effacer l'écran

Aide :

Pour réaliser un beep, écrivez `printf ("%c",0x07);`

7.6 Complément d'exercice

Si l'on arrive au bord de l'écran ($x < 1$, $x > 80$, $y < 1$, $y > 25$), faire en sorte que le curseur passe de l'autre côté de l'écran.

Si $x < 1$ alors $x = 80$

Si $x > 80$ alors $x = 1$

Si $y < 1$ alors $y = 25$

Si $y > 25$ alors $y = 1$

7.7 La cerise sur le gâteau ...

Les touches de direction Haut Bas Droite Gauche sont en fait composées de 2 touches la touche de code Ascii 0 et respectivement les codes Ascii des touches H, P, K, M.

Pour pouvoir utiliser ces touches, il faut donc faire un double switch de la façon suivante :

```
switch (car)
{
    case 0:
        car = getch ();
        switch (car)
        {
            case 'H':
                y --;
                break;
            ...
        }
}
```

Complétez le programme pour qu'il fonctionne aussi avec les touches de direction.

Corrigés des exercices du Chapitre 7

! **Exercice d'application n°1**

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int main ()
{
    clrscr ();
    gotoxy (12,7);
    printf ("Bonjour, c'est moi et j'utilise le C");
    gotoxy (23,18);
    printf ("C'est bien, on peut faire ce que l'on veut");
    getch ();
    return (0);
}
```

! **Exercice d'application n°2**

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int main ()
{
    char car;

    clrscr ();
    gotoxy (1,1);
    printf ("Saisie d'une touche :");
    car = getch ();
    gotoxy (1,3);

    switch (car)
    {
        case 'g':
        case 'G':
            printf ("Gauche\n");
            break;

        case 'd':
        case 'D':
            printf ("Droite\n");
            break;

        case 'h':
        case 'H':
            printf ("Haut\n");
            break;

        case 'b':
        case 'B':
            printf ("Bas\n");
            break;

        default:
            printf ("Rien à faire pour ce cas\n");
            break;
    }
}
```

```
    }

    getch ();
    return (0);
}
```

! Exercice : animation d'un curseur à l'écran

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int main ()
{
    char car;
    int x=0, y=0;
    int sortie=1;

    clrscr ();

    printf ("Saisie de x:\n");
    scanf ("%d",&x);
    printf ("Saisie de y:\n");
    scanf ("%d",&y);

    do
    {
        clrscr ();

        gotoxy (x,y);
        printf ("#");

        car=getch ();
        car = toupper (car);    /* Evite le test minuscule/majuscule */

        switch (car)
        {
            case 'G':
                x--;
                if (x<1)
                    x = 80;
                break;

            case 'D':
                x++;
                if (x>80)
                    x = 1;
                break;

            case 'H':
                y--;
                if (y<1)
                    y = 25;
                break;

            case 'B':
                y++;
                if (y>25)
                    y = 1;
                break;

            case 'S':
```

```
        sortie=0;
        break;

    default:
        printf ("%c",0x07);
        break;
    }
} while (sortie==1);
return (0);
}
```

! **La cerise sur le gâteau ...**

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int main ()
{
    char car;
    int y=10,x=10;
    int sortie = 0;

    clrscr ();
    do
    {
        car=getch ();
        switch (car)
        {
            case 0:
                car=getch();
                switch(car)
                {
                    case 'H':
                        y--;
                        break;
                    case 'P':
                        y++;
                        break;
                    case 'K':
                        x--;
                        break;
                    case 'M':
                        x++;
                        break;
                }
                break;

            case 'S':
            case 's':
                sortie = 1;
                break;
        }

        clrscr ();
        gotoxy (x,y);
        cprintf ("*");

    } while (!sortie);

    return (0);
}
```