

## Bataille navale simplifiée

Adaptée du livre de Jacques Deconchat : "102 Programmes Pour MO6, TO8 et TO9+"

### Le jeu

```

  0  1  2  3  4  5  6  7  8  9
0  .  .  .  *  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .
2  .  .  .  .  .  .  .  .  .  .
3  .  .  .  .  .  *  .  .  .  .
4  .  .  .  *  .  .  .  .  .  .
5  .  .  .  .  .  .  .  .  .  .
6  .  .  .  .  .  .  .  .  .  .
7  .  .  .  .  .  .  .  .  .  .
8  .  .  .  .  .  .  .  .  .  .
9  .  .  .  .  .  .  .  .  .  .
Ligne ? 6          Colonne ? 3
Distance : 6
  Un exemple de Bataille Navale
```

Nous présentons ici un jeu simplifié de la bataille navale. Celui-ci ne comportera qu'un seul navire et le tableau des cases dans lequel peut se déplacer le navire sera limité à 10 cases sur 10 cases.

A chaque coup on devra donner les coordonnées en donnant l'horizontal (H) et la verticale (V). L'ordinateur répondra si le navire est coulé ou dans le cas contraire donnera la distance qui nous sépare du navire.

En fin de partie, on donnera le nombre de coups nécessaires à la destruction du navire.

### Aide pour la programmation

Soit :

d : la distance (int) qui sépare le tir du navire.

x : la position horizontale du navire.

y : la position verticale du navire.

posx\_bat : la position horizontale du tir.

posy\_bat : la position verticale du tir.

La distance qui sépare le navire du tir est calculé par la formule suivante :

$$d = (\text{int}) (\text{sqrt} ((x - \text{posx\_bat}) * (x - \text{posx\_bat}) + (y - \text{posy\_bat}) * (y - \text{posy\_bat})))$$

*N'oubliez pas d'inclure la bibliothèque <math.h>*

### Complément d'informations pour la programmation

En mode texte on dessinera un tableau avec des numéros de 0 à 9 en hauteur et longueur. Les cases en elles-mêmes seront symbolisées par des points.

Dans un premier temps, on indiquera la solution afin de pouvoir contrôler le bon fonctionnement de notre programme.

Dans un second temps, la solution ne sera pas plus donnée et les coups précédemment tirés seront signifiés par des \* au lieu du . .

La fonction atoi permet de transformer une chaîne de caractère en nombre. Pour connaître sa syntaxe consulter l'aide.

## Exemple de solution

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

/* Dessine le plateau de jeu */
void plateau ()
{
    int i=0,j=0;

    clrscr ();
    gotoxy (3,1);
    for (i=0; i<10; i++)
        printf ("%d ",i);

    for (i=0; i<10; i++)
    {
        gotoxy (1,2+i);
        printf ("%d ",i);
        for (j=0; j<10; j++)
            printf (". ");
    }

    gotoxy (1,12);
    printf ("Ligne : ");
    gotoxy (11,12);
    printf ("Colonne : ");
    gotoxy (1,13);
    printf ("Distance : ");
}

/* Saisie un chiffre en position x y */
int saisie_chiffre (int x, int y)
{
    char car;
    char chaine [2];
    int sortie = 1; /* Ok on peut sortir */

    do
    {
        /* On efface la précédente case */
        gotoxy (x,y);
        printf (" ");

        /* On se repositionne pour la saisie */
        gotoxy (x,y);
        car = getch ();
        if ((car<'0') || (car>'9'))
        {
            /* Saisie incorrect : beep */
            sound (1000);
            delay (10);
            nosound ();
            /* On doit recommencer */
            sortie = 0;
        }
        else
            sortie = 1;
    }
}
```

```
while (!sortie);

/* Affichage du caractère saisi */
gotoxy (x,y);
printf ("%c",car);

/* atoi transforme une chaîne de caractère en nombre */
/* transformation de notre caractère en chaîne */
chaîne [0] = car;
chaîne [1] = 0;
return (atoi (chaîne));
}

void main ()
{
    int posx_bat=0, posy_bat=0; /* Position du bateau */
    int x,y; /* Position du tir */
    int nb_coups=0; /* Nombre de coups */
    int d=0; /* Distance */

    randomize ();
    posx_bat = random (10); /* 0 ... 9 */
    posy_bat = random (10);

    plateau ();
    do
    {
        /* Saisie de la colonne */
        x=saisie_chiffre (9,12);

        /* Saisie de la ligne */
        y=saisie_chiffre (21,12);

        /* Nombre de coups */
        nb_coups ++;

        /* Affichage du coup */
        gotoxy ((x+1)*2+1, y+2);
        printf ("");

        /* Effacement des coordonnées */
        gotoxy (9,12);
        printf (" ");
        gotoxy (21,12);
        printf (" ");

        /* Calcul de la distance */
        d = (int) (sqrt (((x-posx_bat)*(x-posx_bat))+((y-posy_bat)*(y-
posy_bat))));

        /* Affichage de la distance */
        gotoxy (12,13);
        printf ("%d",d);
    }
    while (!((x==posx_bat) && (y==posy_bat)));
    /* On recommence tant que la position saisie n'est pas celle du navire */

    gotoxy (1,15);
    printf ("Nombre de coups : %d",nb_coups);
    getch ();
}
```