

TP N°6

Chasse au trésor

Préambule

Nous cherchons à développer un petit jeu, à un joueur, où le but est de rechercher un trésor dans une grille de jeu de taille fixe (20x20 par exemple) :

Grille de jeu

```
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X   X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X   X X X X X X X   X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X   X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X
```

Pour guider le joueur, les cases qui ne contiennent pas le trésor révèlent des indications au joueur. Il peut s'agir de deux types d'indications :

- Une indication de position du trésor : indique si le trésor est en haut, en bas, à gauche ou à droite (les combinaisons du type en haut à gauche sont possible).

Entrez la prochaine colonne a jouer ? 5

Entrez la prochaine ligne a jouer ? 7

Direction

Le tresor se trouve en bas à droite

- Une indication de distance du trésor : indique la distance, en cases, qui nous sépare du trésor.

Entrez la prochaine colonne a jouer ? 8

Entrez la prochaine ligne a jouer ? 9

| Distance |

```
-----  
Le tresor se trouve a une distance d'environ 8 cases.
```

Le jeu se termine lorsque le joueur a trouvé le trésor.

1. Gestion de la grille

Type de case et constante

Avant de commencer, déclarez un ensemble de constantes permettant de définir les différentes valeurs que peut prendre une case de la grille, à savoir :

- *VIDE* : la case ne contient pas le trésor et n’a pas été découverte (vaut 0).
- *DECOUVERTE* : la case est découverte mais ne contient pas le trésor (vaut 1).
- *TRESOR_CACHE* : la case contient le trésor mais n’a pas été découverte (vaut 2).
- *TRESOR_DECOUVERT* : la case est découverte et elle contient le trésor (vaut 3).

De plus, vous pouvez déclarer une constante contenant la taille de la grille de jeu, ici 20 cases.

Fonction d’initialisation de la grille

Écrivez une fonction permettant d’initialiser chaque case de la grille, passée en paramètre, à *EVide*. La déclaration de la fonction est la suivante :

```
void InitialiserGrille (...);
```

Fonction d’affichage de la grille

Écrivez une fonction permettant d’afficher chaque case d’une grille passée en paramètre. Elle devra afficher :

- 'X' si la case vaut *VIDE*.
- ' ' si la case vaut *DECOUVERTE*.
- 'X' si la case vaut *TRESOR_CACHE*.
- '\$' si la case vaut *TRESOR_DECOUVERT*.

La déclaration de la fonction est la suivante :

```
void AfficherGrille (...);
```

TESTER VOTRE PROGRAMME

2. Gestion des cases

Fonction de demande des coordonnées

Écrivez une fonction permettant de demander à l'utilisateur la ligne et la colonne qu'il souhaite jouer. On prendra soin de vérifier que les valeurs saisies sont supérieures à 0 et qu'elles ne dépassent pas la taille de la grille. La déclaration de la fonction est la suivante :

```
void DemanderLigneEtColonne (int* ligne, int* colonne);
```

Attention : ici il est nécessaire de renvoyer à la fois la ligne et la colonne choisies par l'utilisateur. Pour cela on utilise un passage de paramètres par pointeur. Dans ce type de passage, ligne et colonne contiennent les adresses des variables de ligne et de colonne de la fonction appelante. Pour accéder à ces variables dans la fonction à écrire il faudra donc mettre une étoile (*) devant le nom des variables.

Fonction d'initialisation du trésor

Écrivez une fonction permettant d'initialiser la position du trésor dans la grille. Pour cela, il est nécessaire de tirer aléatoirement les coordonnées x et y de la case trésor et de définir la valeur de cette case à `TRESOR_CACHE` :

```
void CacherTrésor (...);
```

Remarque : en langage C la génération d'un nombre aléatoire se fait grâce à la fonction `rand()`. Pour générer un nombre aléatoire entre 0 et N exclu on utilise la syntaxe suivante :

```
int valeurAleatoire;  
valeurAleatoire = rand()%N;
```

Pour éviter que le tirage soit toujours identique (il s'agit en fait d'une simple suite mathématique U_n) il est nécessaire d'initialiser le générateur de nombres aléatoires (avec le paramètre n de la suite, aussi appelé graine) comme suit :

```
#include <time.h>

int main ()
{
    ...
    srand (time(NULL));
    ...
}
```

TESTER VOTRE PROGRAMME

3. Gestion des fonctions d'aides

Fonction de d'affichage de la distance

Écrivez une fonction permettant d'afficher la distance qui sépare le joueur du trésor. Pour cela on utilisera la formule suivante :

$$distance = |x2 - x1| + |y2 - y1|$$

Pour calculer la valeur absolue, on utilisera la fonction *abs ()* du langage C. La déclaration de la fonction est la suivante :

```
void AfficherDistance (int x1, int y1, int x2, int y2);
```

Fonction de d'affichage des informations de direction

Écrivez une fonction permettant d'afficher les informations donnant la direction du trésor. Celles-ci doivent-être :

- En haut.
- A gauche.
- En bas.
- A droite.

La combinaison de plusieurs directions est possible. Par exemple on pourra afficher : « *en haut à gauche* » si le trésor se trouve dans cette direction. La déclaration de la fonction est la suivante :

```
void AfficherInformations (int x1, int y1, int x2, int y2);
```

TESTER VOTRE PROGRAMME

4. Programme principal

Écrivez une fonction principal *main()* permettant de créer la boucle de jeu présentée en préambule de ce document.