

Bataille navale

Devoir 2 - Algorithmique et bases de la programmation

Jeudi 16 novembre 2023

Vous devez écrire un programme permettant de jouer au jeu "bataille navale" dans une version solo, en mode console. L'ordinateur place les navires, puis l'utilisateur joue pour trouver, c'est-à-dire couler, tous les navires en un minimum de coups.

Le jeu se déroule sur une grille de 10x10 cellules, avec un navire de cinq cellules, un navire de quatre cellules, trois navires de trois cellules et quatre navires de deux cellules.

1 Cahier des charges et implémentation

Le programme doit implémenter les fonctionnalités suivantes :

- placement des bateaux en tirant au hasard leur orientation et leur position : les bateaux doivent être positionnés, sur la grille, soit verticalement soit horizontalement, **ils ne doivent pas se chevaucher**.
- jeu : l'utilisateur tire sur des cellules de la grille, dont il saisit les coordonnées, jusqu'à ce que tous les bateaux soient coulés, ou qu'il choisisse d'abandonner.

Les colonnes de la grille doivent être numérotées de 0 à 9, et les lignes de A à J. **La cellule en haut à gauche doit être la cellule A0.**

Au début de la partie, le programme doit demander au joueur s'il souhaite que les cases testées mais vides, c'est-à-dire cibles de tirs dans l'eau, soient affichées sur la grille. Bien entendu, les cases occupées par un bateau, qui ont déjà été découvertes, doivent être affichées quel que soit le choix du joueur.

A chaque tour de jeu, la grille doit être affichée, et le joueur doit saisir **en une seule fois** les coordonnées de la cellule visée, par exemple B5. De plus, à chaque tour de jeu, le joueur doit pouvoir abandonner **proprement** la partie. A la fin de la partie, si le joueur n'a pas abandonné, le programme doit afficher le nombre de coups utilisés pour couler tous les bateaux.

Un symbole macrodéfini doit permettre l'activation, ou la désactivation, d'un mode débogage. En mode débogage, les cellules occupées par un bateau et pas encore découvertes doivent aussi être affichées (différemment des cellules occupées découvertes).

Attention. Le mode débogage ne doit pas être accessible à un utilisateur "lambda" : il ne doit pas être une option d'un menu. Le mode débogage sera

toujours désactivé dans l'exécutable fourni à un joueur "lambda". Ce mode est réservé aux développeurs : son activation permet ... de faciliter le débogage du programme.

Vous n'avez pas le droit d'utiliser des variables globales.

Dans ce programme, toutes les saisies doivent être **fiabilisées** : si l'utilisateur saisit n'importe quoi, votre programme ne doit pas planter, et l'utilisateur doit être invité à effectuer une nouvelle saisie. De même, si l'utilisateur saisit, par exemple, une chaîne de caractères trop longue, il ne doit pas y avoir de débordement de tableau.

Vous avez le droit d'utiliser les fonctions définies dans les corrigés des exercices du cours, et toutes les fonctions existantes dans la bibliothèque standard, sauf les fonctions `malloc()` et `calloc()` qui permettent d'allouer de la mémoire dans le tas (les fonctions `malloc()` et `calloc()` seront vues dans le cours "Programmation avancée" lors du prochain semestre).

Question bonus. Le programme informe l'utilisateur quand la dernière cellule d'un bateau est touchée : le bateau est alors "touché coulé".

2 Rendu

Le programme est à écrire en langage C. Vous devez rendre sur Moodle un fichier source d'extension `.c` portant votre nom. Par exemple, Jean Dupont doit rendre un fichier nommé `dupont.c`.

De plus, vous devez indiquer, dans un commentaire placé au début du fichier source :

- votre nom et votre prénom ;
- ce qui fonctionne et ce qui ne fonctionne pas.

3 Critères d'évaluation

Liste non exhaustive :

- compilation sans warning en utilisant les options `-std=c99`, `-Wall` et `-Wextra` ;
- qualité du code et des algorithmes, factorisation du code, **découpage en fonctions** ;
- indentation, commentaires, noms des variables et des fonctions explicites ;
- signatures¹ des fonctions : pertinence du choix et du typage des paramètres, et de la valeur de retour ;

¹Chaque prototype de fonction doit être justifié avec un bloc de commentaires qui indique le rôle de la fonction, ainsi que ceux des paramètres à transmettre et de la valeur de retour.

- ergonomie du programme.

Tout programme rendu qui ne compile pas, ne sera pas corrigé.

Annexe : affichage et saisie d'une variable de type `size_t`

Le "length modifier"² `z` a été ajouté dans la norme C99 du langage C et doit être utilisé pour afficher ou saisir un `size_t` avec `printf()` ou `scanf()`.

Exemple :

```
size_t var;  
printf(" > ");  
scanf("%zu", &var);  
printf("var = %zu\n", var);
```

²Le cours présente uniquement deux "length modifier" : il s'agit des "length modifier" `l` et `h`.