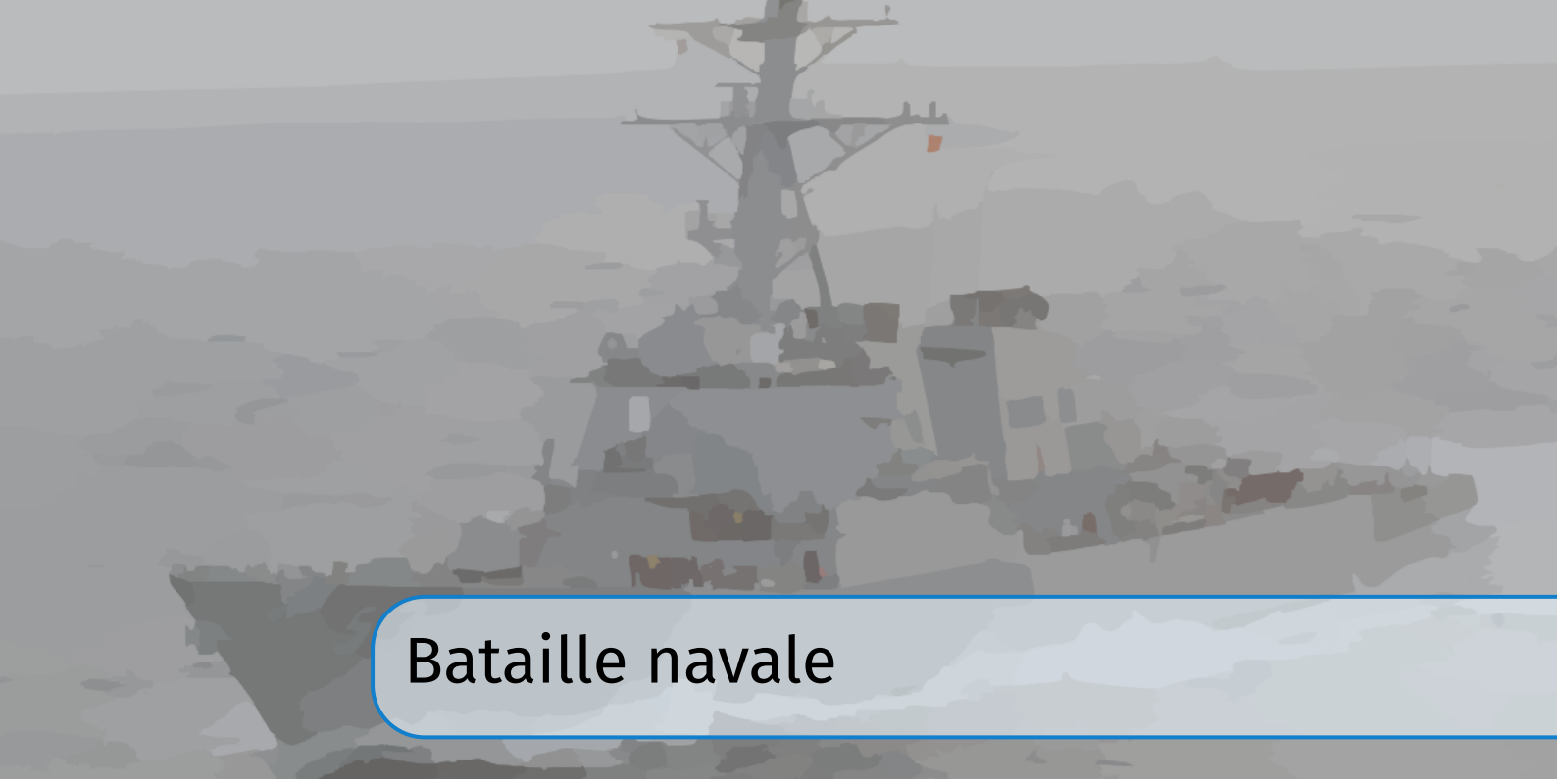

Bataille navale

If debugging is the process of removing bugs, then programming must be the process of putting them in.

Edsger W. Dijkstra

Objectif

Implémenter un jeu de bataille navale mettant en œuvre la plupart des concepts de programmation précédemment rencontrés.

0.1 Présentation

Dans le jeu de la bataille navale, deux joueurs s'affrontent. Chacun possède une flotte composée de cinq navires : 1 porte-avion (5 cases), 1 croiseur (4 cases), 1 contre-torpilleur (3 cases), 1 sous-marin (3 cases) et 1 torpilleur (2 cases). Au début du jeu, les navires de chaque joueur sont placés sur leur grille personnelle comportant 10×10 cases. Les grilles sont numérotées verticalement de A à J et horizontalement de 1 à 10 comme le montre la figure 1.1. Chacun leur tour, les joueurs vont tirer sur une case adverse avec l'objectif de couler le maximum de navires chez l'adversaire. Le jeu se déroule en 100 coups et le gagnant est celui qui conserve au moins un navire en fin de partie, ou celui qui en possède le plus, si les deux ont encore des navires en état à terme.

0.2 Implémentation

Le jeu de bataille navale à développer se joue en monoposte et va opposer un joueur humain à l'ordinateur. À la différence de la version classique, la position

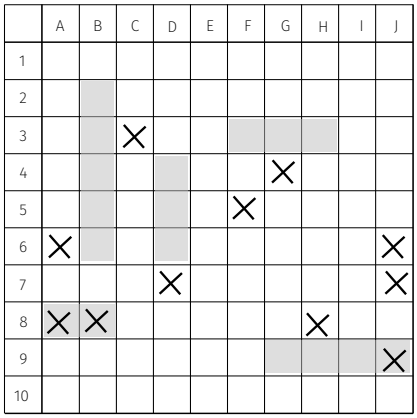

Figure 1: Exemple de grille d'un des joueurs. Les cases grisées indiquent les navires du joueur et les croix renseignent les cases sur lesquelles son adversaire a tiré. Le joueur suivra le succès de ses propres tirs sur une grille distincte.

des navires peut évoluer en cours de jeu (dans un premier temps, vous pouvez laisser de côté cette fonctionnalité si vous la considérez difficile à mettre en œuvre).

Contraintes de l'implémentation

Le placement sera effectué de manière aléatoire pour le joueur "ordinateur" et au choix, aléatoirement ou non pour le joueur "humain".

Les joueurs ne peuvent jouer qu'une fois chacun à leur tour, effectuer un tir sur une case adverse et éventuellement déplacer tout ou partie de leurs navires. À tout instant, le joueur humain visualise sa grille personnelle et l'état de sa flotte, ainsi que la grille de son adversaire (vide au départ) et faisant au fur et à mesure apparaître les coups qui ont été portés avec succès.

Placement et déplacement

Le placement et le déplacement doivent respecter les contraintes suivantes :

- chaque navire est en totalité dans la grille ;
- le placement d'un navire s'effectue en indiquant la case de positionnement de sa proue ainsi que son orientation (NORD, EST, OUEST ou SUD) – la poupe du navire sera donc orientée en sens inverse ;
- le déplacement d'un navire ne peut se faire qu'en déplaçant sa proue d'une case au maximum vers l'avant, à tribord ou à bâbord – dans ces deux derniers cas, son orientation s'en trouve modifiée.

Interface textuelle du jeu

L'implémentation de l'interface textuelle peut se faire à l'aide de la bibliothèque *ncurses* disponible nativement sur les distributions Linux.

Livrable

L'arborescence du projet comportera au minimum :

- un fichier `README` présent à la racine, et décrivant succinctement le projet ainsi que les commandes à effectuer pour installer et exécuter le programme ;
- le fichier `Makefile` permettant de construire automatiquement le programme ;
- les répertoires `include` et `src` comportant le(s) différents fichier(s) source(s) `.c` et `.h` ;
- les répertoires `bin`, `lib` et `doc` comportant respectivement le programme final, les bibliothèques statiques et dynamiques rassemblant les modules que vous considérerez comme partageables, ainsi que la documentation.

La qualité structurelle et la lisibilité du code (emploi de macros, de structures, modularité du code, commentaires, etc.) respectera les règles de codage GNU : https://www.gnu.org/prep/standards/standards.pdf.