

---

## TP3 - INF111-01-02 Programmation orientée-objet(hors programme) (A2016)

---

TP à réaliser en équipe de deux et à rendre le 1er Décembre à 23h55

---

Mathieu Nayrolles

8 novembre 2016

### 1 BATAILLE NAVALE

La bataille navale, appelée aussi touché-coulé, est un jeu de société dans lequel deux joueurs doivent placer des navires sur une grille tenue secrète et tenter de toucher les navires adverses. Le gagnant est celui qui parvient à torpiller complètement les navires de l'adversaire avant que tous les siens ne le soient. Le principe du jeu de bataille navale semble trouver son origine dans le jeu français L'Attaque lors de la Première Guerre mondiale, on a aussi trouvé des liens de parenté avec le jeu de E. I. Horseman en 1890 (Baslinda) et on dit que des officiers russes y auraient joué antérieurement à la première guerre. La première version commerciale du jeu fut publiée en 1931 par la StarexNovelty Co. sous le nom de Salvo. Ce jeu est devenu populaire lors de son apparition en 1943 dans les publications américaines de divertissement de laMilton Bradley Company qui l'exploita sous la forme papier jusqu'en 1967, où elle sortit un jeu de plateau, puis en réalisa une version électronique en 1977.

#### 1.1 LISTE DES NAVIRES

Chaque joueur possède les mêmes navires, dont le nombre et le type dépendent des règles du jeu choisies.

Une disposition peut ainsi comporter :

- 1 porte-avions (5 cases)
- 1 croiseur (4 cases)
- 1 contre-torpilleurs (3 cases)
- 1 sous-marin (3 cases)
- 1 torpilleur (2 cases)

## 1.2 GRILLE DE JEU

Qu'elle soit en version électronique ou non, la grille de jeu est toujours la même, numérotée de 1 à 10 horizontalement et d'A à J verticalement.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

TABLE 1.1 – Une grille de jeu avec les navires placés

Conventionnellement, les joueurs placent des pions blancs sur la grille lorsque les coordonnées n'ont pas touché de bateau adverse, et rouge lorsqu'une touche a été faite.

## 2 RÈGLES

La bataille navale oppose deux joueurs qui s'affrontent. Chacun a une flotte composée de 5 bateaux, qui sont, en général, les suivants : 1 porte-avion (5 cases), 1 croiseur (4 cases), 1 contre torpilleur (3 cases), 1 sous-marin (3 cases), 1 torpilleur (2 cases). Au début du jeu, chaque joueur place ses bateaux sur sa grille. Celle-ci est toujours numérotée d'A à J verticalement et de 1 à 10 horizontalement. Un à un, les joueurs vont “tirer” sur une case de l'adversaire. Par exemple B.3 ou encore H.8. Le but est donc de couler les bateaux adverses. En général, les jeux de société prévoient des pions blancs pour les tirs dans l'eau (donc qui ne touchent aucun bateau adverse) et des pions rouges pour les “touché”. Au fur et à mesure, il faut mettre les pions sur sa propre grille afin de se souvenir de nos tirs passés.

## 3 ÉNONCÉ DU TP

Votre objectif pour ce travail pratique est de créer un joueur artificiel capable de jouer contre vous à la bataille navale de manière graphique.

### 3.1 ALGORITHMES

Le joueur artificiel va avoir deux modes de difficulté : Facile et Normale.

### 3.1.1 DIFFICULTÉ FACILE

Dans le mode facile, le joueur artificiel placera ses navires au hasard sur la carte et effectuera des tirs sur la grille de son adversaire (vous) au hasard jusqu'à ce que l'un des deux joueurs perde. Un joueur perd la partie lorsque tous ses navires sont coulés.

### 3.1.2 DIFFICULTÉ NORMAL

Dans le mode de difficulté normal, les navires sont placés au hasard et les tirs sont effectués au hasard jusqu'à ce qu'un navire soit touché. Une fois qu'un navire est touché, il est possible de parcourir les cases au NORD, à l'EST, au SUD et à l'OUEST de la case courante afin de tirer sur le reste du navire<sup>1</sup>.

Une implémentation simple de cette stratégie consiste à créer une pile dynamique des cibles potentielles. Initialement, l'algorithme commence en mode Chasse et tir au hasard. Lorsqu'un navire est touché, l'algorithme passe en mode Cible. Après un touché, les quatre cases (ou moins si l'on se trouve sur un bord / angle) qui entourent la case courante sont empilées dans la pile des cibles potentielles.

Les cases ne sont ajoutées à la pile des cibles potentielles que si elles n'ont pas encore été ciblées par un tir.

Lorsque le mode Cible est activé, on dépile une cible potentielle de la pile et on tire dessus. Si le tir touche un navire, alors les cases adjacentes sont ajoutées à la pile. L'algorithme reste en mode Cible tant qu'il reste des cibles potentielles dans la pile et qu'il reste des navires adverses. Si la pile est vide et que la partie n'est pas finie, alors l'algorithme repasse en mode Chasse et tir au hasard.

Les figures qui suivent illustrent une partie.

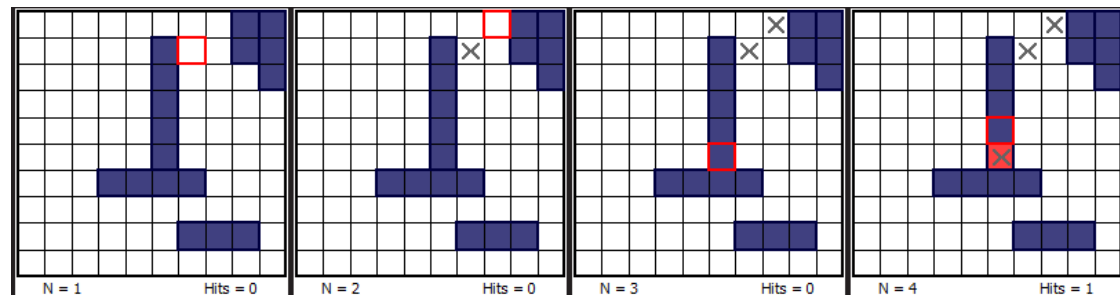


FIGURE 3.1 – Initialement, l'algorithme commence en mode Chasse et tir au hasard. Au tour 3, il touche un navire et le mode Cible est activé. Les quatre cases (NORD, SUD, EST, OUEST) sont ajoutées à la pile des cibles potentielles. Au tour 4 l'algorithme tire au nord du dernier touché et touche à nouveau un navire. Les cases N, E et O de ce nouveau touché sont ajoutées à la pile. La case SUD n'est pas ajoutée car l'algorithme a déjà tiré dessus.

1. Algorithme Nick Berry.

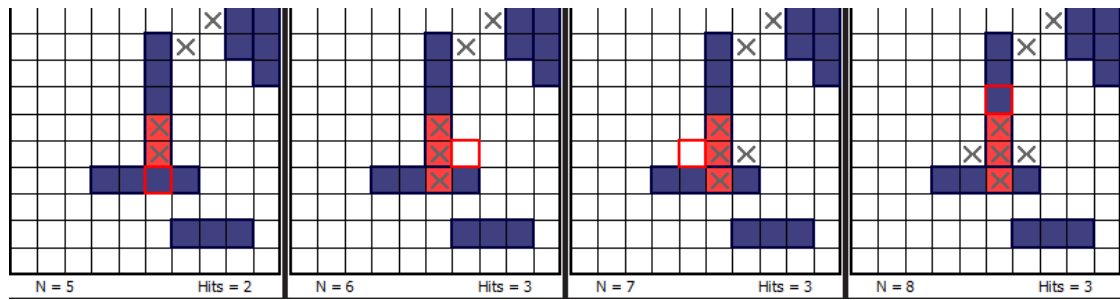


FIGURE 3.2 – Au tour 5, l’algorithme touche encore, on ajoute donc les cases adjacentes qui n’ont pas encore été visitées. Les tours 6 et 7 ne touchent pas.

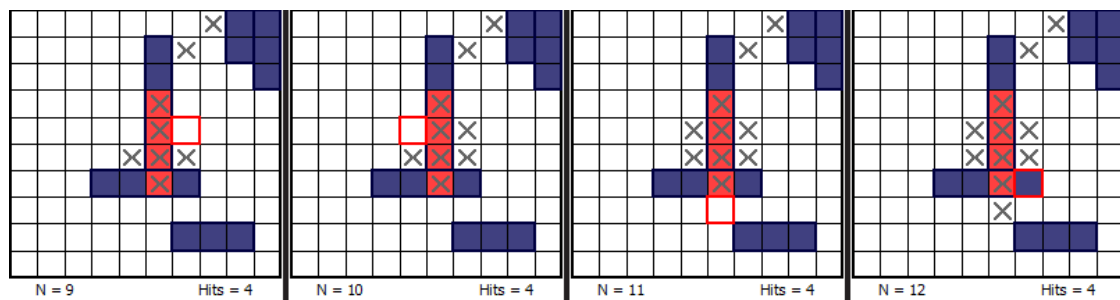


FIGURE 3.3 – Les tours 9 et 10 montrent comment l’algorithme test et élimine les cases des deux côtés du navire. Au tour 12, l’algorithme découvre un nouveau navire.

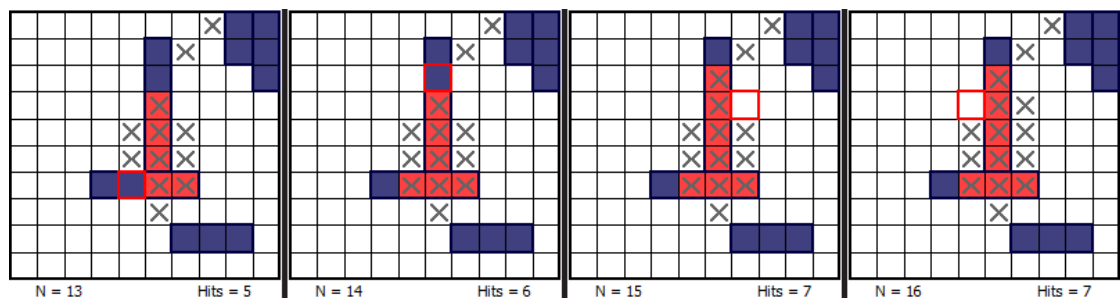


FIGURE 3.4 – L’exploration de ce groupe de navires continue.

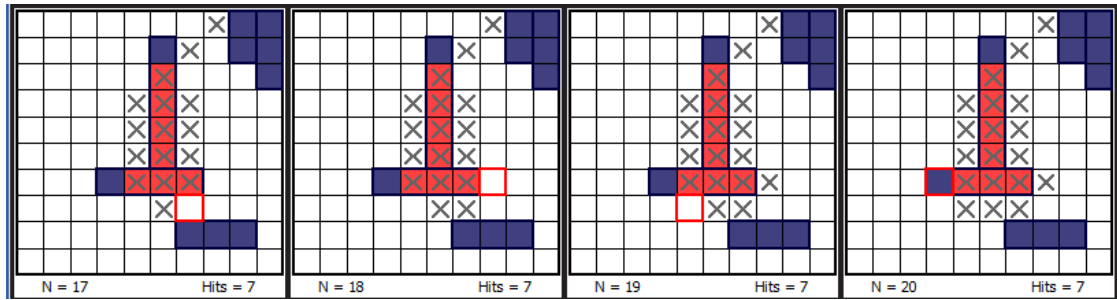


FIGURE 3.5 – Pour être certain, nous devons tirer sur toutes les cases adjacentes à une case qui a donné un touché.

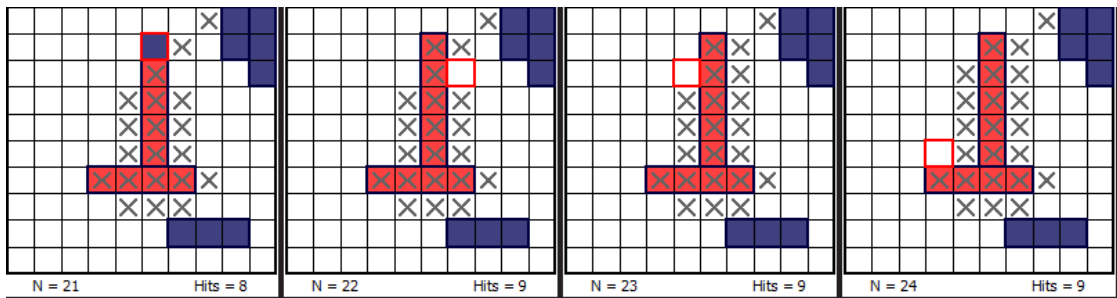


FIGURE 3.6 – Au tour 21, nous avons un nouveau touché.

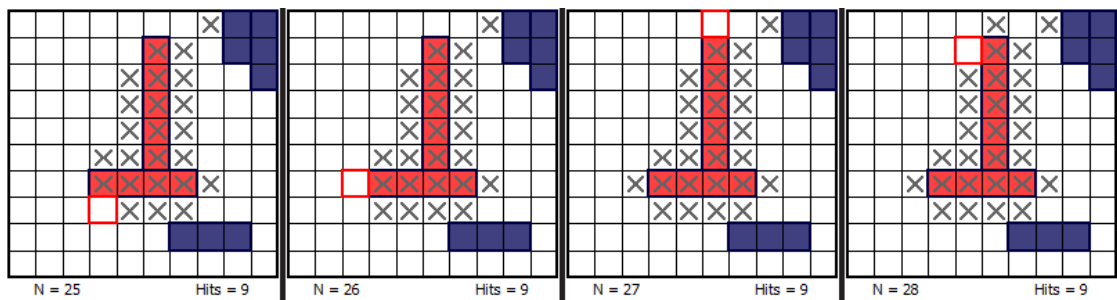


FIGURE 3.7 – Au tour 28, la dernière cible potentielle a été dépilée de la pile, l'algorithme repasse donc en mode Chasse.

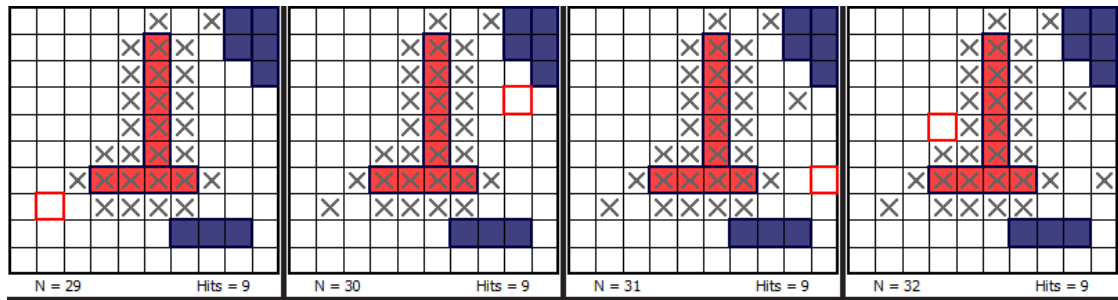


FIGURE 3.8 – Aucun touché ici.

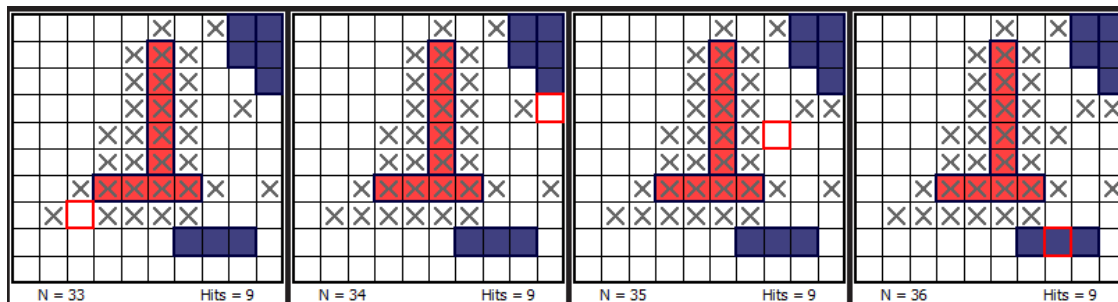


FIGURE 3.9 – Au tour 36, nous avons un touché, l'algorithme repasse en mode Cible.

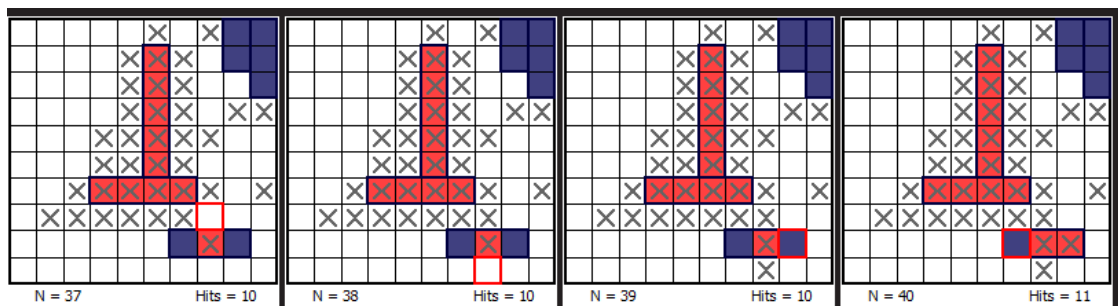


FIGURE 3.10 – Au tour 40, le croiseur est coulé.

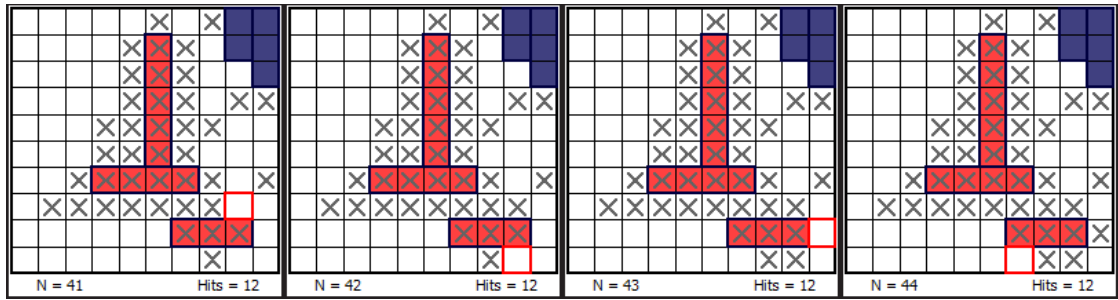


FIGURE 3.11 – Certaines cases ont déjà été visitées ce qui accélère la recherche.

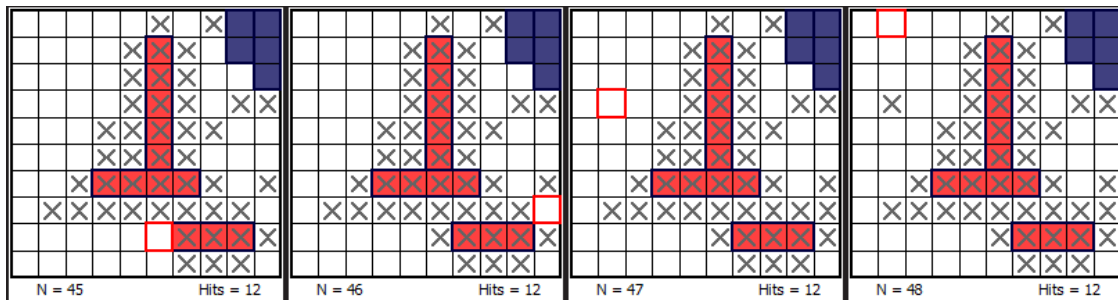


FIGURE 3.12 – Le raté du tour 45 fait repasser l'algorithme en mode Chasse.

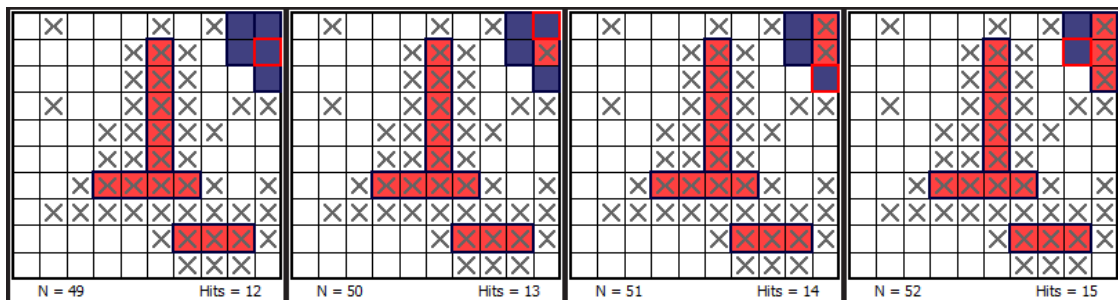


FIGURE 3.13 – Nous avons un nouveau touché au tour 49

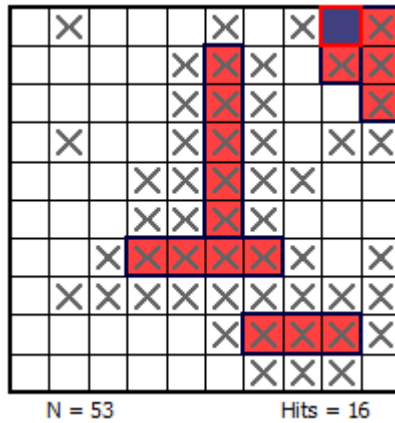


FIGURE 3.14 – Finalement, les navires sont coulés au tour 53.

### 3.1.3 CONTRAINTES D'IMPLEMENTATION

Au niveau de l'implémentation des différentes classes :

- Vous devez avoir une interface `IJoueur` qui contient les services suivants
  - `tire(IJoueur cible)`
  - `reçoitTire(Position position)`
- Vous devez avoir une classe abstraite `AJoueur` qui implémente l'interface `IJoueur` et implémente l'algorithme "normal". En plus, cette classe abstraite contient tout les services nécessaire au positionnement des bateaux (`placeNavire(Navire navire)`). L'algorithme "normal" est factorisé au niveau de la classe abstraite car le joueur humain va aussi l'utiliser (voir prochaine section).
- La classe `AJoueur` est étendue par une classe `JoueurArtificiel` et une classe `JoueurHumain`. Chacune des classes spécialises `AJoueur` de manière différente.
- La classe `Navire` contient toutes les informations nécessaire pour modéliser un navire
  - Liste dynamique de `Position`. Une position est une case de la carte (i.e. A-1). Une position, en plus de la position en lettre et en chiffre contient un boolean `touché` qui indique si cette position a déjà été visée par un tir.
  - boolean `estCoulé()` qui retourne vrai si toute les positions ont `touché == true`.
  - Le nom du navire (i.e. porte-avions, ...)

## 3.2 L'INTERFACE À RÉALISER

La figure ci-dessous présente l'interface graphique à réaliser.



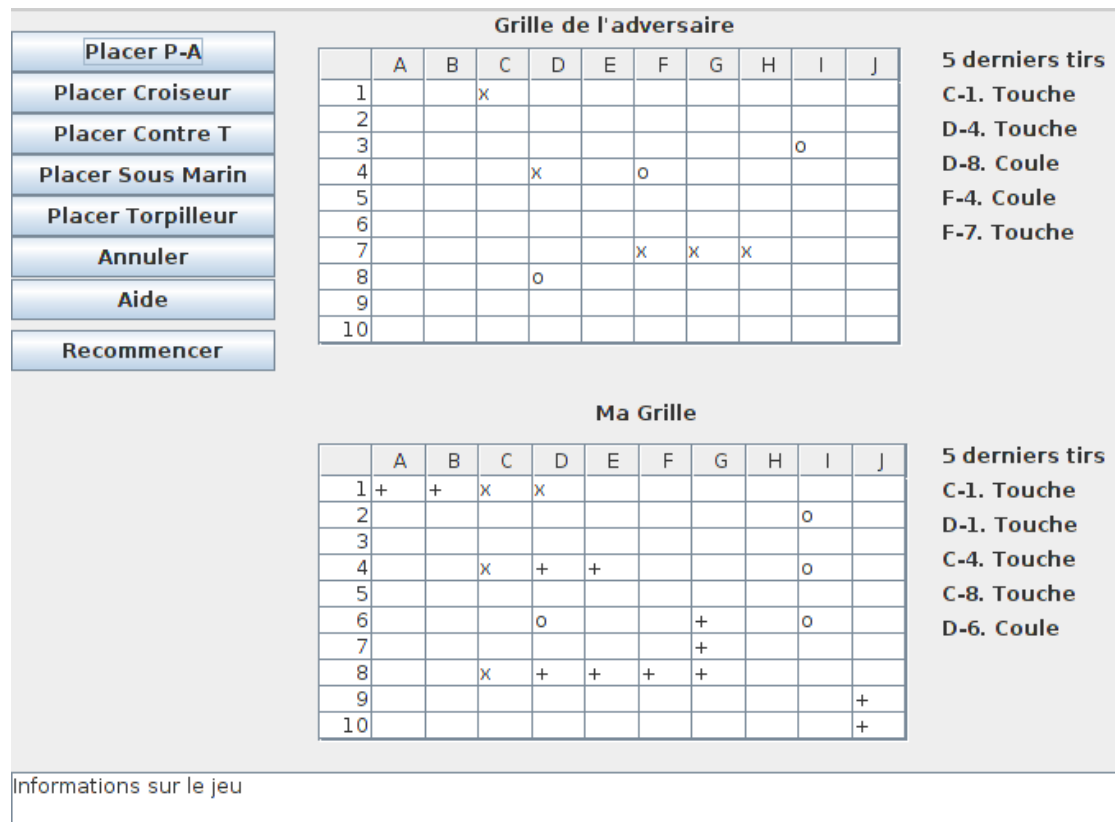


FIGURE 3.15 – Interface de base

- Placer : *Place un*
- Annuler : Annule le dernier coup du joueur humain.
- Aide : Suggère une case sur laquelle tirer pour le joueur humain grâce au même algorithme que le joueur artificiel
- Recommencer : Recommence une nouvelle partie.
- 5 Derniers coups : Affiche les cinq derniers coups.
- Zone de texte en bas : Affiche les informations sur la partie. (Choisissez une case pour faire feu, Vous avez gagné, ect...).
- x, o et + : Les x sont des tirs touchés, les o sont des tirs à l'eau et les + sont vos bateaux non touchés.

### 3.2.1 COMPORTEMENT DE L'INTERFACE

Lorsque la partie commence, les deux grilles sont vides et le joueur humain doit placer ses navires. Pour placer un navire, le joueur humain clique sur l'un des boutons de placement. Lorsque l'un des boutons est choisi, les autres se grisent comme sur la figure 3.16. Le joueur humain, doit cliquer sur la case de début du navire et ensuite sur la case de fin. Bien entendu, le nombre de cases entre le début et la fin du navire doit correspondre au nombre de cases

du navire choisi. De plus, les navires ne peuvent se chevaucher et ils ne peuvent pas être en diagonale.

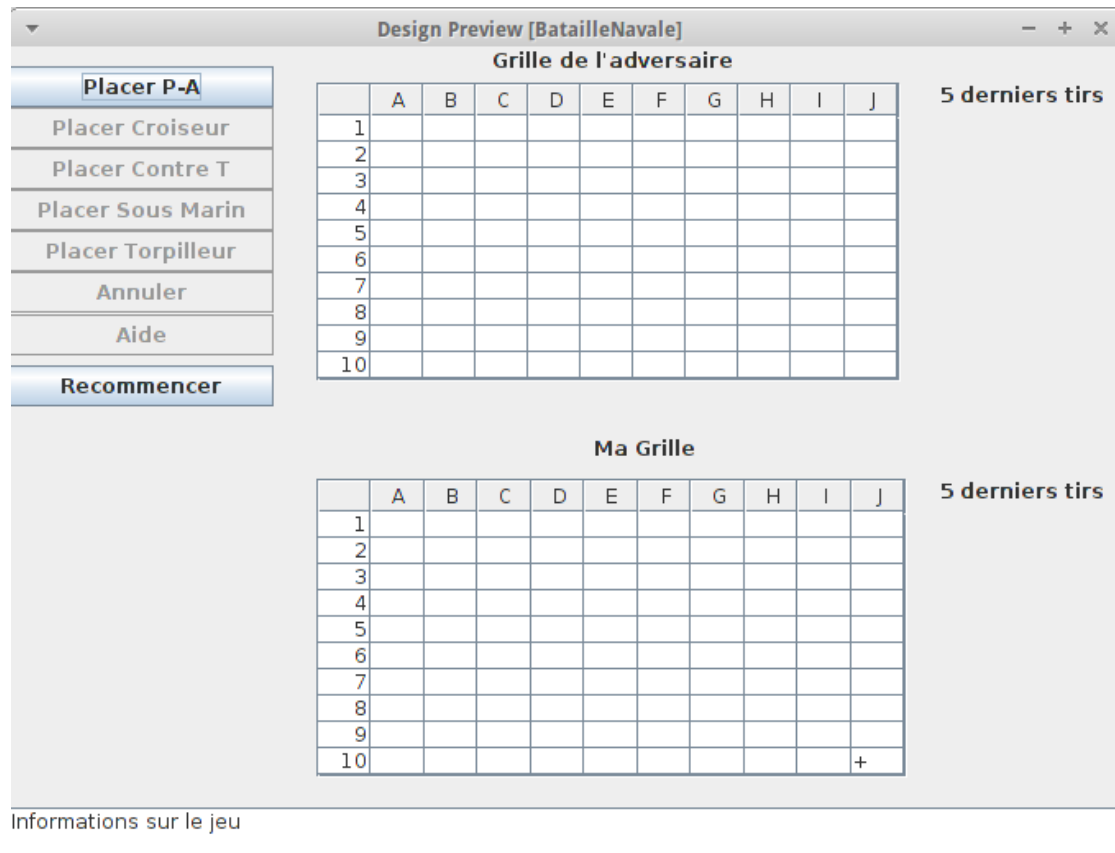


FIGURE 3.16 – Placement d'un bateau.

Le joueur humain choisit place ensuite son second navire. Le porte-avion étant déjà placé, le bouton est grisé.

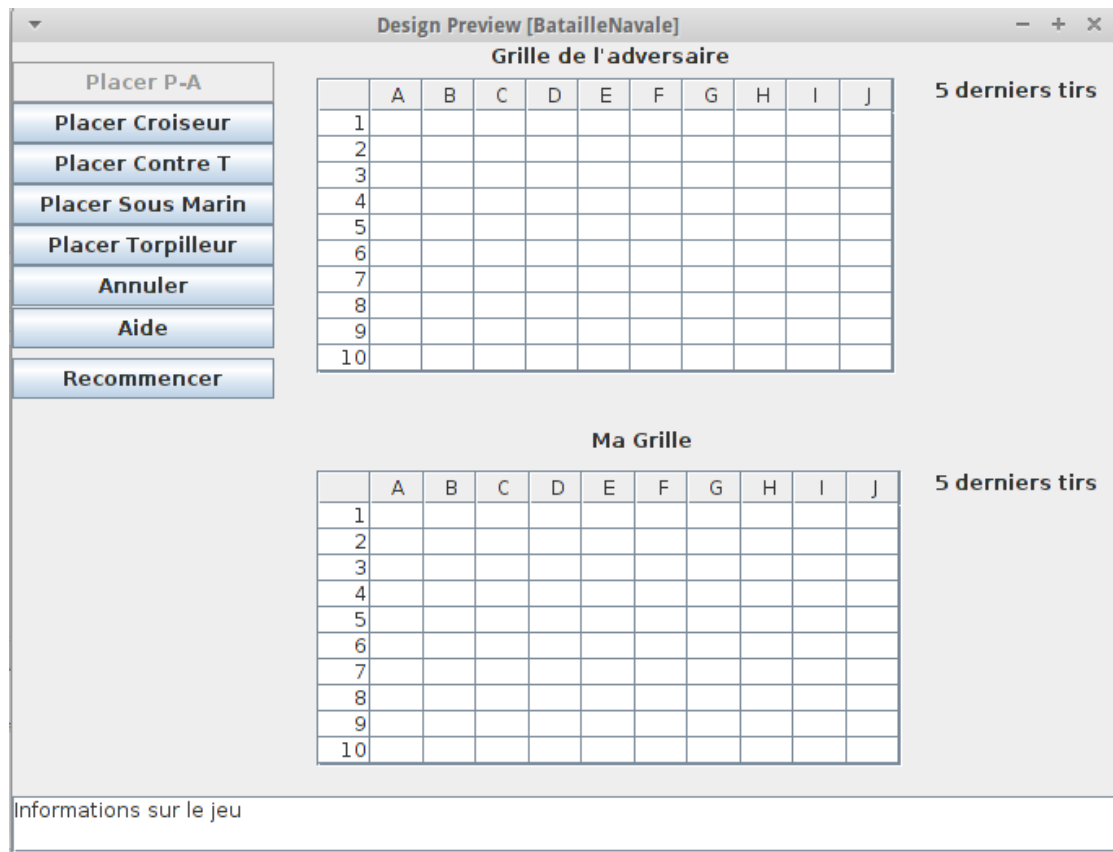


FIGURE 3.17 – Placement d'un second bateau.

Lorsque les navires sont tous placés, la partie commence et le joueur humain fait feu en cliquant sur une case de la grille de son adversaire. Après le clic, la case se transforme en x ou o si c'est un touché ou un coulé. Pareillement, la liste des cinq derniers coups est mise à jour.

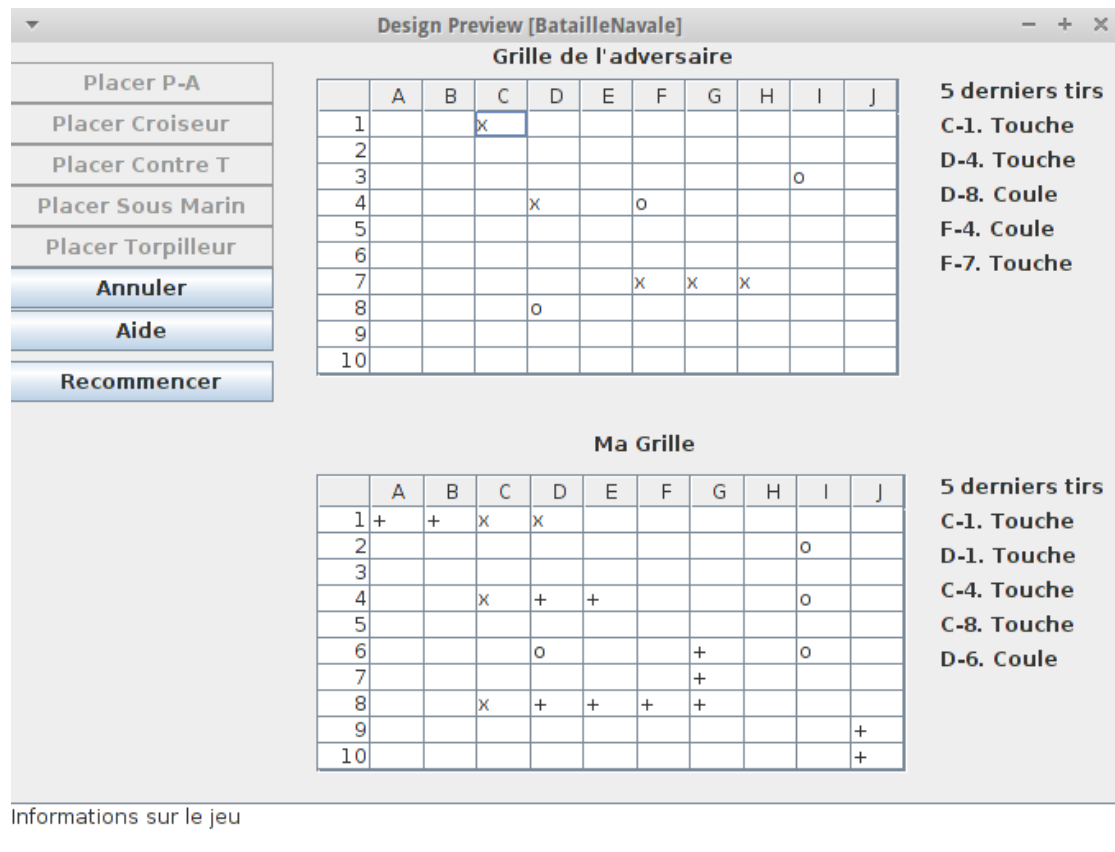


FIGURE 3.18 – Action de tir.

### 3.2.2 IMPLÉMENTATION DE L'INTERFACE

Voici les règles d'implémentation à respecter :

- Les boutons sont créés grâce à une classe personnalisée qui hérite de JButton.
- Les grilles peuvent être, au choix :
  - Des JTable
  - Une suite de JButton
  - une suite de JTextArea
- Les boutons de gauche sont dans un JPanel
- Chaque grille est dans un JPanel
- Chaque "5 derniers coups" est dans un JPanel et implémente le comportement observateur.
- Les coups des joueurs sont stockés dans des files dynamiques ce qui permet d'annuler un coup.
- Le bouton "Aide" qui suggère une case sur laquelle tirer au joueur humain fait appel au même algorithme (ensemble de sous programmes) que le joueur artificiel.
- Les comportements suivants s'affichent dans la zone texte en bas :

- Statistiques (nombre de parties joués / gagnés. Nombre de coups moyen pour une victoire)
- Case déjà ciblée
- Placement correct des navires
- Fin de partie
- Aucun point n'est attribué sur la "beauté" de l'interface graphique. Si votre interface respecte les normes ci-dessus et l'organisation générale de l'interface décrite ici, vous ne serez pas pénalisé.

## 4 CORRECTIONS

Les normes de programmation établies pour les TP1 et TP2 sont toujours en vigueur. L'exécution compte pour 60% et la qualité du code pour 40%.