

Bataille navale

Équipe enseignante PC EUR/AMER

1 Présentation

Le but de ce projet est de réaliser un jeu de bataille navale, aussi appelé « touché/coulé ». Le principe en est le suivant : chaque joueur dispose d'un certain nombre de bateaux, posés sur un plateau. Dans la version traditionnelle du jeu, ces bateaux sont fixes. À tour de rôle, chaque joueur va tirer un projectile sur le plateau adverse, dont il ne connaît pas le contenu initial.

Le joueur indique les coordonnées qu'il vise (par exemple, « A5 » pour désigner la cinquième colonne de la première ligne, ou « B4 » pour désigner la quatrième colonne de la deuxième ligne). Le joueur adverse lui répond alors le résultat de son tir : « dans l'eau » (le projectile n'a touché aucun bateau), « touché » (le projectile a touché un bateau, mais celui-ci flotte encore), ou « coulé » (le projectile a achevé de couler le bateau).

Un bateau est coulé lorsque toutes les cases qu'il occupe ont été touchées. Le gagnant est le premier joueur à avoir coulé tous les navires de son adversaire.

2 Fonctionnement

Ce projet offre plusieurs possibilités d'évolutions, ou branches, chacune offrant plusieurs niveaux de complexité. Les groupes pourront choisir d'avancer plus ou moins chacune de ces branches, voire de proposer leurs propres améliorations. *Pour atteindre la moyenne, il faut avoir mis en place le niveau deux pour au moins deux des branches (et au moins le niveau un pour les autres).*

2.1 Les bases

Deux classes sont requises :

- la classe Bateau représentera un bateau actuellement en jeu. Elle sera capable de renseigner sur la longueur et la position du bateau, et d'indiquer son état (indemne, touché, ou coulé).

Une méthode « public boolean tir(int x, int y) » servira à prévenir un bateau qu'on vient de tirer. Elle devra renvoyer vrai si le bateau a été touché, faux si le tir a manqué le bateau. Dans le cas où le bateau est touché, la méthode modifie son état.

- la classe Plateau représentera le plateau d'un joueur. Un plateau contiendra tous les bateaux de son joueur, et mémorisera l'emplacement des tirs

reçus. Elle pourra être affichée directement, ou transmise à un gestionnaire d'affichage.

Au début du jeu, chaque participant devra placer ses bateaux, en indiquant la coordonnée de départ et la direction (verticale ou horizontale). Une fois ceci fait, chacun des joueurs tirera à tour de rôle, jusqu'à ce que l'un des joueurs remporte la partie.

2.2 Branche « intelligence artificielle »

Une partie peut être jouée soit entre humains, soit entre humain et ordinateur. Plusieurs niveaux de complexité sont possibles :

1. Le jeu ne permet de jouer qu'une seule des deux alternatives (entre humains ou entre humain et ordinateur, au choix du groupe). L'IA peut dans ce cas être très basique (par exemple, choisir au hasard une case du plateau sur laquelle tirer, tant que la case n'a pas été choisie précédemment)
2. Les deux cas peuvent être joués. L'utilisateur est donc invité, au lancement du programme, à décider s'il veut jouer contre un autre humain ou contre l'ordinateur. Là encore, l'IA peut rester très basique.
3. Plusieurs niveaux d'intelligence artificielle sont disponibles (par exemple, une IA basique tirant toujours aléatoirement, et une autre qui, si elle vient de toucher un bateau, va ensuite tirer à proximité jusqu'à l'avoir coulé).

2.3 Branche « nombre de joueurs »

La partie traditionnelle se joue à deux. Cependant, on peut envisager des variantes à plus de joueurs :

1. Jeu de base, deux joueurs (humain ou ordinateur) s'affrontent.
2. Trois joueurs ou plus s'affrontent. L'utilisateur doit alors indiquer le nombre de joueurs au lancement du programme. L'ordre de jeu est déterminé, de même que les cibles visées (le joueur 2 tire toujours sur le joueur 1, le joueur 3 tire toujours sur le joueur 2, etc.)
3. Trois joueurs ou plus s'affrontent, mais les cibles visées ne sont plus déterminées : chaque joueur, au début de son tour, doit choisir quelle sera sa cible parmi les autres joueurs.

2.4 Branche « forme de plateau »

Le jeu traditionnel se joue sur un plateau rectangulaire, de taille fixe. Pour simplifier, nous considérons que tous les joueurs ont des plateaux identiques. Cependant, des variantes sont possibles :

1. Le plateau est rectangulaire et de taille fixe. Le nombre de bateaux est fixé également.
2. Le plateau est rectangulaire, mais l'utilisateur choisit sa taille au lancement de la partie. Il indique alors combien de bateaux seront utilisés, en précisant les tailles.
3. Le plateau est rectangulaire, mais l'utilisateur choisit sa taille au lancement de la partie. Le nombre et la taille des bateaux est alors déterminé automatiquement.
4. L'utilisateur peut choisir des formes de plateaux différentes (triangulaire, circulaire...)

2.5 Branche « affichage »

Plusieurs manières de gérer l'affichage sont envisagées. Des points bonus seront décernés aux groupes implémentant plusieurs rendus graphiques et permettant à l'utilisateur de choisir le rendu qu'il désire.

1. Affichage simple dans le terminal. Une fois que le joueur a tiré, le programme lui affiche seulement le résultat de son action (« touché », « coulé » ou « dans l'eau »).
2. Affichage graphique. Comme pour le TP sur le jeu de la vie, une classe fournie permet d'afficher l'état actuel du plateau à partir d'un tableau d'entier à deux dimensions (se référer à la javadoc).
3. Affichage avancé dans le terminal : le plateau entier est affiché, de cette manière :

```

+---+---+---+
| | |O| | |
+---+---+---+
| | |T|T| |
+---+---+---+
|T|O| | | |
+---+---+---+

```

2.6 Branche « saisie »

Plusieurs manières de saisir les commandes peuvent être utilisées :

1. le joueur indique séparément le numéro de la ligne et le numéro de la colonne, par lecture directe de nombres tapés au clavier.
2. pour les groupes ayant implémenté l'interface graphique pour la visualisation, l'utilisateur clique sur les cases (se référer à la javadoc fournie).

3. le joueur indique l'endroit où il vise en une seule étape, par une chaîne comprenant des lettres suivies de chiffres. Le programme étudie alors cette chaîne pour déterminer quelle case est visée.