

Compte-rendu du jeu Bataille Navale

Quentin SAUVAGE

26 avril 2017

Chapitre 1

Fonctionnement du programme

Le jeu débute sur une fenêtre accueillant le joueur et lui proposant de jouer contre un joueur humain ou contre l'IA. Après avoir choisi, une nouvelle fenêtre s'ouvre, contenant les grilles respectives de chaque joueur. Avant que ces grilles ne s'affichent, le jeu place aléatoirement les bateaux de chaque joueurs, en veillant à ce que chaque partie d'un bateau soient alignées. Après cela, les joueurs choisissent à tour de rôle une case sur la grille adverse. Le joueur peut alors savoir s'il a touché ou coulé un certain bateau, ou s'il a tiré dans le vide, grâce au panneau situé au milieu du jeu, permettant de garder un historique de la partie. Il est également averti des effets de son tir sur la grille adverse : Une croix est affichée s'il a raté, sinon, la case se remplit de noir. Pour que l'IA commence son tour, le joueur doit cliquer sur sa propre grille. A la fin de la partie, le jeu demande au joueur s'il veut recommencer une partie ou quitter.

Chapitre 2

Structure du programme

Voici le rôle de chacune des classes du programme :

Package controler :

- AbstractControler définit les méthodes devant être utilisées par GrilleControler.
- GrilleControler vérifie les actions du joueur en appelant les méthodes adéquates du Model.

Package initialisation :

- Main permet de lancer l'application.
- Debut permet de définir les paramètres de la partie : Présence ou non d'IA.

Package model :

- AbstractModel définit les méthodes et attributs de Plateau.
- Bateau est une Enum définissant tous les types de bateaux, ainsi que la case "vide". Cette classe est utilisée pour indiquer quel type de bateau a été touché.
- CaseJeu associe une valeur entière à un Bateau.
- IA hérite de Joueur et peut jouer sans intervention humaine, comme définit plus bas.
- Joueur possède une matrice de CaseJeu représentant sa grille. Il a ainsi accès au type de Bateau associée à une case.
- Observable est une interface définissant les méthodes permettant au Model de gérer ses Observers.
- Plateau place les bateaux de chaque joueurs sur leur grille en faisant attention à ce qu'aucun ne se chevauchent.

Package view : - Case représente graphiquement une case du plateau, pouvant être un bateau, ou vide.

- GuiPlateau contient tous les éléments graphiques du jeu.
- Observer définit les méthodes mettant à jour les Objet Observables.

Chapitre 3

Fonctionnement de l'IA

3.1 Premier fonctionnement de l'IA

L'IA tire aléatoirement sur la grille, en évitant de tirer là où elle a déjà tiré (en vérifiant dans les quatre directions). Une fois qu'elle a touché un bateau, elle tire tout autour de ce bateau jusqu'à ce que toutes les cases autour aient été choisies. Si aucun bateau ne se trouvait autour, elle recommence à tirer aléatoirement, sinon elle tire de nouveau tout autour du nouveau bateau touché.

Chapitre 4

Difficultés principales du projet

La principale difficulté fut de placer les bateaux en veillant à ce qu'il soit sur une même ligne, sans chevaucher un bateau déjà existant. J'ai donc dû passer en paramètre la taille du bateau à placer afin de voir s'il était possible de le placer entièrement une fois qu'une case était choisie. Pour cela, j'ai dû vérifier qu'il existait bien une direction dans laquelle il y avait n cases vides d'affilée, n étant la taille du bateau à placer. Il a également fallu différencier si un bateau était simplement touché, ou coulé. J'ai enfin également rendu possible l'accès au type du bateau touché, afin de rendre le jeu plus compréhensible et les parties moins longues.