

Projet Bataille navale JAVA

1) Les différentes classes du projet

Le projet est constitué de deux packages. Le package client et le package serveur. Le package client contient simplement le code et les classes permettant de connecter un client. Le package serveur contient le code du serveur qui crée les plateaux de jeux associés aux deux clients. Seulement deux clients peuvent lancer une partie avec l'architecture utilisée. Le package serveur contient une classe grille qui contient toutes les méthodes nécessaires au fonctionnement du jeu. La classe ThreadChat contient le fonctionnement de la partie c'est dans les méthodes run() et start() du ThreadChat qu'on appelle les méthodes de la classe grille.

2) Déroulement d'une partie

Les deux joueurs se connectent dans la console puis le joueur 1 va devoir rentrer les coordonnées de ses bateaux comme indiqué dans la console. Par exemple pour placer un porte-avion sur la ligne 0 en commençant par la colonne 0 il faudra saisir '001'. Le programme ne gère malheureusement pas l'ensemble des mauvaises saisies de l'utilisateur. Pour les gérer, à chaque fois qu'on demande à l'utilisateur une saisie, il faudrait faire une boucle while et des conditions pour vérifier la validité de la saisie selon le format imposé.

Une fois que le joueur 1 a placé ses bateaux c'est au tour du joueur 2.

A chaque fois qu'un joueur place un bateau le threadchat appelle des méthodes de la classe grille chargé de vérifier les placements dans le cas où le format est respecté. Par exemple si un bateau est collé à un autre bateau, la console affichera un message demandant à l'utilisateur de refaire une nouvelle saisie, de même lorsque le bateau ne rentre pas dans la grille par exemple. Une fois que les deux grilles ont été remplies le serveur utilise la méthode run() du threadchat. Les joueurs ont un premier affichage de leur grille et la partie peut commencer. A chaque tour des joueurs, le joueur doit saisir le coup qu'il veut jouer à nouveau sous la forme d'un format imposé. Les exceptions ne sont encore une fois pas traitées par conséquent, l'utilisateur peut faire facilement planter le programme avec une saisie non conforme. A chaque fois qu'un utilisateur joue un coup, des méthodes pour déterminer si le coup touche, coule ou rate sont appelées ainsi qu'une méthode pour déterminer s'il y a un vainqueur.

3) La classe grille

La classe grille contient toutes les méthodes nécessaires au bon déroulement de la partie. Une grille de jeu est une matrice de taille 10*10. La méthode grille permet d'initialiser un plateau de jeu pour un joueur. Il s'agit d'une matrice de 0 de taille 10*10.

La méthode placeC() permet de placer un bateau sur une colonne. Cette méthode doit être appelée après avoir vérifié que le placement était correct. La méthode placeL() fait la même chose mais pour le placement en ligne. Les méthodes VerifieL() et VerifieC() vérifient simplement que le placement demandé ne dépasse pas la grille.

Les méthodes `VerifiePlacementC()` et `VerifiePlacementL()` vérifient que le placement est valide dans le sens où le nouveau bateau ne touche pas un ancien bateau. La méthode `affiche`, affiche simplement la grille. La méthode `Joue()` permet de jouer un coup, un coup est un tableau de deux valeurs : la première valeur est la ligne, la seconde est la colonne. La méthode `check()` vérifie la grille est une grille perdante (tout les bateaux sont coulés). La méthode `bateauEnLigne()` permet de vérifier si un bateau est placé en ligne. La méthode `EstCoule()` permet de vérifier si un coup coule un bateau. Cette méthode à besoin de savoir si un bateau est placé en ligne ou en colonne. Puis elle parcourt tout le bateau à partir de l'indice donnée (à gauche et à droite pour un bateau en ligne, en haut et en bas pour un bateau en colonne), elle s'arrête quand elle tombe sur l'indice 0 car cela marque une fin de bateau. Un indice 1 indique que le bateau n'est pas coulé. Cette méthode est appelée lorsqu'un bateau a été touché et que donc il est susceptible d'être coulé.

4) problème de terminaison et de correction

Etant donnée, que certaine partie du projet demande d'utiliser des boucles et des conditions l'existence de problèmes de terminaison et de correction n'est pas impossible. Je n'ai en effet pas eu le temps de vérifier que les différentes méthodes se termine bien et qu'elles font bien ce qu'on attend d'elle. Par conséquent il pourrait y avoir des exceptions non traitées dans les méthodes de la classe grille. Par ailleurs, le bon fonctionnement du programme dépend de l'appel des méthodes dans le bon ordre, par exemple on ne peut pas vérifier qu'il n'y a pas de bateau à l'emplacement demandé avant d'avoir vérifié qu'on n'essaye pas de placer le bateau en partit en dehors de la grille.

5) Pistes d'améliorations

- vérifié la correction et la terminaison des méthodes pour être sûr qu'on n'oublie pas d'exceptions
- modifié l'architecture de la classe serveur pour permettre à plusieurs clients de se connecter et de jouer en même temps
- gérer toutes les exceptions de saisie des utilisateurs à l'aide de boucle while et de conditions.