Université de la Manouba  
Ecole Nationale des Sciences de l’informatique  
  
   
  
  
 Sujet :

|  |
| --- |
| Implémentation du Jeux Bataille Navale |

Réalisée Par :

***Achraf kriaa***

**1-Présentation générale du sujet :**  
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 couler (c'est-à-dire toucher toutes les cases) tous les navires de l'adversaire avant que tous les siens ne le soient.

Régles du jeu :

La bataille navale oppose deux joueurs. Chaque joueur dispose de deux grilles carrées de côté 10, dont les lignes sont numérotées de 1 à 10 et les colonnes de A à J, ainsi que d'une flotte composée de quelques bateaux d'une à cinq cases de long.

L'une des grilles représente la zone contenant sa propre flotte. Au début du jeu, chaque joueur place ses bateaux sur sa grille, en s'assurant que deux bateaux ne sont pas adjacents. L'autre représente la zone adverse, où il cherchera à couler les bateaux de son adversaire.

Chaque joueur, à son tour, annonce une case (par exemple « B6 »), et son adversaire lui répond si le tir tombe à l'eau ou au contraire s'il touche un bateau. Dans ce dernier cas, il annonce « touché » s'il reste des cases intactes au bateau ciblé, et « touché-coulé » si non.

**2-presentation générale de la solution :**  
 **2.1-archeticture du programme :**  
**Le jeu est formé principalement par les classes suivantes :Cellule,Bateau,Plateau.**  
 **2.2-Détails de la solution :**  
**Pour mieux comprendre l’utilités des classes déclarés ci-dessus on doit faire une petite récapitulation sur les attributs et les méthodes de chaque classe.**

***A-Class Cellule :***  
La **cellule** représentera l'entité de base de notre plateau de jeu. A l'intérieur de cette cellule, pourront être stockés des navires ou des cases de mer.  
***A.1-Les attributs:***  
Protected:

- navire : Bateau (null)

- estVisitee : boolean (false): qui permettra de garder une trace de l'envoi de la torpille ou non dans cette case.

***A.2-Les méthodes :***  
5 méthodes :

+ estVisitee() : boolean : indique si la cellule a été visitée ou non (si une torpille a été ou non envoyée).

+ faitPartiebateau() : boolean : indique si la cellule contient un bateau ou pas.

+ mettreBateau(Bateau b) : pour placer un bateau dans cette cellule.

+ estCoulee() : boolean : interroge la classe bateau afin de savoir si le navire est coulé.

+ visite() permet de positionner l'attribut estVisitée à true pour indiquer que la cellule a été visitée.  
  
  
  
***B-Class Bateau :***  
Cette classe permet de définir un bateau(navire).

***B.1-Les attributs :***  
**Private :**

**- ligneDépart, colonneDépart : int**

**- direction : boolean**

**- taille : int**  
***B.2-Les méthodes:***  
**+ estCoule()** : boolean: parcourt toutes les cellules du navire. Si elle trouve qu'une des cellules n'est pas visitée, le navire n'est alors pas coulé et on retourne la valeur false.

Si toutes les cellules du navire ont été visitées (torpillées), on retourne la valeur true. Comme les bateaux peuvent être positionnés verticalement ou horizontalement, deux cas sont nécessaires.

***C-Plateau :***  
**C’est une classe qui hérite du classe Cellule et du classe Bateau.(on peut dire c’est le champ de bataille) .**  
***C.1-Les attributs :***  
**Private :**

**- grille [ ] [ ] : Cellule**

**- Bateaux [ ] : Bateau**

**4 constantes :**

**+ TAILLE\_PLATEAU : 10**

**+ HORIZONTAL : true**

**+ VERTICAL : false**

**+ placement [ ] : {5,4,3,2,2,1,1}**

***C.2-Les méthodes :***  
**+plateau() :** **Le constructeur de la classe Plateau effectue deux actions distinctes, elle crée la grille puis elle positionne des navires sur le plateau.**

**+void peupleNavires() : on place les navires 1 par un , leur taille est donnée par la constante(5,4,3,2,2,1,1).**

**+void peupleGrille() : on crée les cellules du plateau de jeu une par une .**

**+Cellule getCellule(int ligne, int colonne) : retourne la cellule aux ligne et colonne passées en paramètres ou null si la cellule n'existe pas.**

**+bool estLibre(int ligne, int colonne,bool direction, int taille) : retourne vrai ou faux si on peut placer ou pas un bateau dans cette position.**

**+void peuple(int ligneDepart, int colonneDepart, bool direction, int taille, Bateau bateau) :**

**on positionne le bateau à une position donnée.**

**+ bool faitPartieDeBateau(int ligne, int colonne) : retourne si la cellule aux coordonnées ligne, colonne fait partie ou non d'un bateau .**

**+ bool estVisitee(int ligne, int colonne) : retourne si la cellule aux coordonnées ligne, colonne a été visitée (torpillée).**

**+bool estCoulee(int ligne, int colonne) : retourne si la cellule aux coordonnées ligne, colonne a été coulée.**

**+void visite(int ligne, int colonne) : visite (lance une torpille) à cellule à la position ligne, colonne.**

**+ bool estDetruitPartout() : vrai si tous les bâteaux ont été touchés.**

***D – main :***  
**- Elle commence par créer et afficher le plateau.**

**• une boucle est exécutée jusqu’à ce que tous les bateaux soient coulés. C’est la méthode estDetruitPartout() de la classe Plateau qui est utilisée.**

**• Dans cette même boucle, on demande à l’utilisateur d’envoyer une torpille et la cellule ciblée sera marquée comme "visitée .**

**• Si la cellule fait partie d’un navire, nous informons alors le joueur que le navire est endommagé ou détruit. C’est la méthode estCoulée() de la classe Plateau qui est utilisée.**

**• si la cellule ne fait pas partie d’un navire, on en informe l’utilisateur**

**3.Exemple :**   
