TD8 : définition d'interfaces dans le cadre d'une conception MCV

Bataille navale

Comme son nom l'indique, la bataille navale est un jeu consistant à torpiller les navires de son adversaire. La flotte de chaque joueur est placée aléatoirement au début du jeu par l'application. Elle est composée :

* 1 porte-avion (5 cases consécutives, non diagonal)
* 2 croiseurs (4 cases consécutives, non diagonal)
* 2 sous-marins (3 cases consécutives)
* 2 torpilleurs (2 cases consécutives).

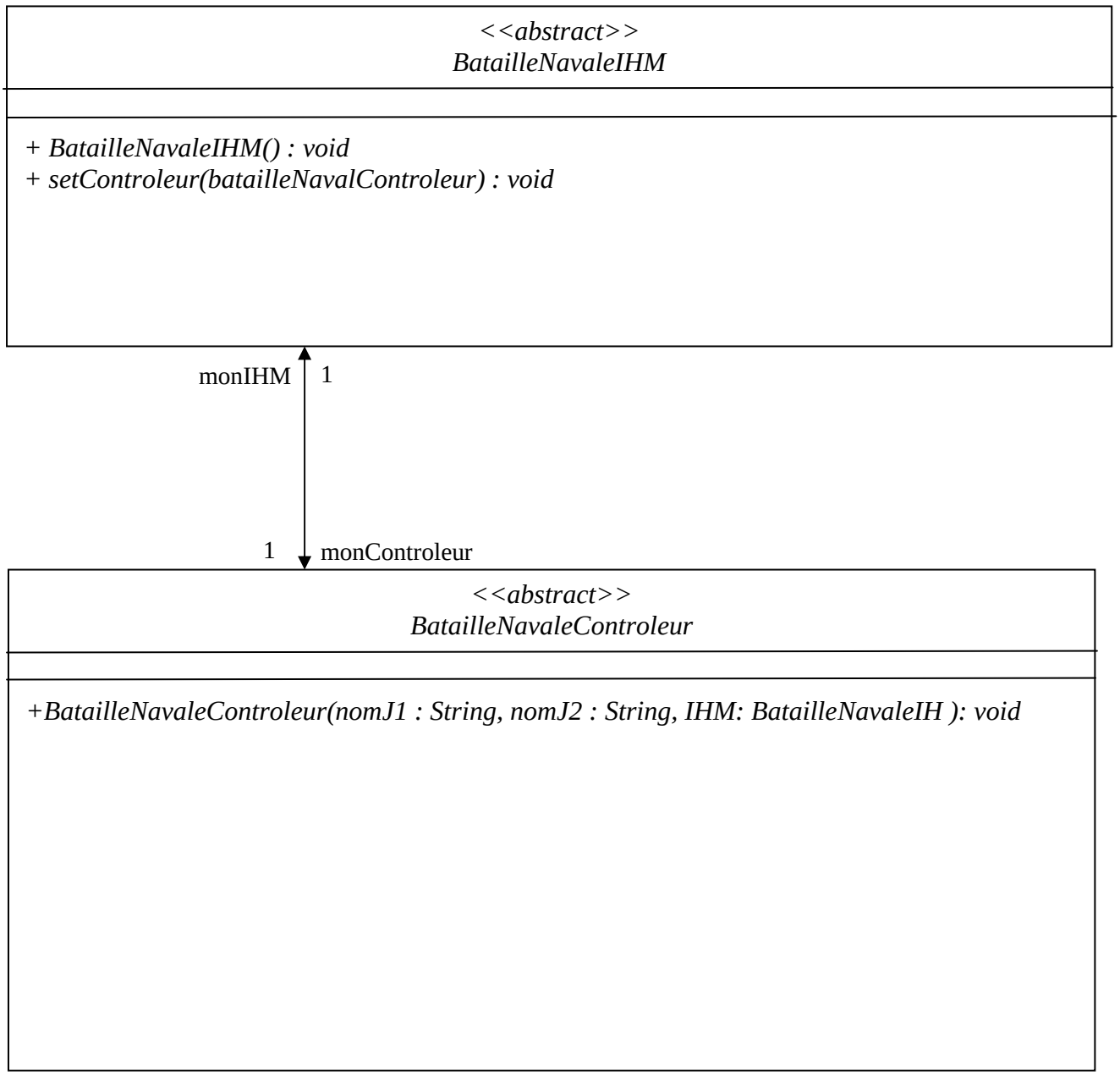
Ensuite, tour à tour, les joueurs lancent virtuellement des torpilles sur l'ennemi en indiquant les coordonnées d'un tir (par exemple B4). L'application indique si l'un de ses bateaux a été atteint ou pas. Le gagnant est celui qui parvient à torpiller complètement les navires de l'adversaire avant que tous les siens ne le soient.

La taille des grilles de jeu est toujours la même 10x10 : numérotée de 1 à 10 horizontalement et de A à J verticalement.

Chaque joueur voit une grille correspondant à la connaissance qu'il a acquise sur la flotte de son adversaire : une case sans pion est une case non torpillée, un pion blanc indique une torpille qui n'a pas touché un navire ; un pion rouge indique une touche.

1. Listez les événements externes et les événements résultats.
2. Complétez la classe abstraite `BatailleNavaleControleur` et l'interface `BatailleNavaleIHM` : listez les méthodes nécessaires au bon fonctionnement du logiciel.
3. Donnez la signature des méthodes.
4. Etablir le diagramme de classes de la partie gestion du jeu.

Remarque : l'interface `BatailleNavaleControleur` est la façade de la partie gestion du jeu.


<<abstract>>
BatailleNavaleIHM
+ BatailleNavaleIHM() : void
+ setControleur(batailleNavalControleur) : void
monIHM
1
1
monControleur
<<abstract>>
BatailleNavaleControleur
+BatailleNavaleControleur(nomJ1 : String, nomJ2 : String, IHM: BatailleNavaleIH): void