

TP2 - Rapport

1 Exercice 1

On a créé un document `testBoard.java` et on tape:

```
public class testBoard {  
    public static void exercice1() {  
        Board ziqi = new Board("ziqi");  
        Board dada = new Board("dada",5);  
        ziqi.print();  
        dada.print();  
    }  
}
```

Les résultats sont ci-dessous:

Navires:											Frappes:										
	A	B	C	D	E	F	G	H	I	J		A	B	C	D	E	F	G	H	I	J
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

Navires:						Frappes:					
	A	B	C	D	E		A	B	C	D	E
0	0
1	1
2	2
3	3
4	4

2 Exercice 2

Les résultats de l'exercice 2 sont dans le dossier `ensta/model/ship`.

3 Exercice 3

3.1 Pistes de réflexion

- **Les indices de position:** dans notre grille, les indices commencent à 0.
- **Si la valeur (position + longueur) d'un navire mène en dehors de la grille de jeu**, on ne peut pas le mettre dans la grille.
- **Si deux navires se chevauchent**, on ne peut pas mettre le deuxième bateau dans la grille.

3.2 Test

On teste les fonctions comme ci-dessous:

```
public static void exercice3() {
    Board ziqi = new Board("ziqi");
    AbstractShip b1 = new Battleship();
    AbstractShip b2 = new Destroyer();
    AbstractShip b3 = new Carrier();
    Coords p1 = new Coords(4,1);
    Coords p2 = new Coords(7,3);
    Coords p3 = new Coords(5,1);
    Coords p4 = new Coords(8,4);
    ziqi.putShip(b1, p1);
    ziqi.putShip(b2, p2);
    ziqi.putShip(b3, p3);
    ziqi.putShip(b3, p4);
    ziqi.print();
}
```

Le bateau3 et le bateau1 se chevauche, donc le met du bateau3 est échoué, et le met du bateau3 sur la position p4 est échoué aussi, car la valeur est en dehors de la grille de jeu.

```
On ne peut pas mettre le bateau: ( Carrier , est), dans la position: (5, 1).
On ne peut pas mettre le bateau: ( Carrier , est), dans la position: (8, 4).
Navires:
  A B C D E F G H I J
0 . . . . . . . . . .
1 . . . . B B B B . .
2 . . . . . . . . . .
3 . . . . . . D D . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .

Frappes:
  A B C D E F G H I J
0 . . . . . . . . . .
1 . . . . . . . . . .
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .
```

4 Exercice 4

On écrit le document `testPlayer.java`.

```
public class testPlayer {
    public static void main(String args[]) {
        exercice4();
    }
    public static void exercice4() {
        Board ziqi = new Board("ziqi");
        Board dada = new Board("dada");
        List<AbstractShip> ships = new ArrayList<>();
        ships.add(new Destroyer());
        ships.add(new Submarine());
        ships.add(new Submarine());
        ships.add(new Battleship());
        ships.add(new Carrier());
        Player p1 = new Player(ziqi,dada, ships);
        p1.putShips();
    }
}
```

```
}

}
```

il faut taper la position au format « A1 n », « B4 w », « D3 s », si la format n'est pas bon, on a une indication:

```
placer 1 : Destroyer(2)
A0
Format incorrect! Entrez la position sous forme 'A0 north'
A0 south
```

Entre chaque saisie, on affiche l'état du Board :

```
placer 1 : Destroyer(2)
A0
Format incorrect! Entrez la position sous forme 'A0 north'
A0 south
Navires:
  A B C D E F G H I J
0 D . . . . . . . . .
1 D . . . . . . . . .
2 . . . . . . . . .
3 . . . . . . . . .
4 . . . . . . . . .
5 . . . . . . . . .
6 . . . . . . . . .
7 . . . . . . . . .
8 . . . . . . . . .
9 . . . . . . . . .

placer 2 : Submarine(3)
A1 south
```

Si la valeur d'un navire mène en dehors de la grille de jeu ou deux navires se chevauchent, il faut redemander de l'entrée de la position:

```
placer 2 : Submarine(3)
A1 south
On ne peut pas mettre le bateau: ( Submarine , sud), dans la position: (0, 1).
On ne peut pas le met dans le board, essayez un autre:
B3 east
Navires:
  A B C D E F G H I J
0 D . . . . . . . . .
1 D . . . . . . . . .
2 . . . . . . . . .
3 . S S S . . . . .
4 . . . . . . . . .
5 . . . . . . . . .
6 . . . . . . . . .
7 . . . . . . . . .
8 . . . . . . . . .
9 . . . . . . . . .
```

5 Exercice 5

On ajoute une fonction `exercice5` dans le document `testBoard.java` :

```
public static void exercice5() {
```

```

Board ziqi = new Board("ziqi");
AbstractShip b1 = new Battleship();
AbstractShip b2 = new Destroyer();
Coords p1 = new Coords(4,1);
Coords p2 = new Coords(7,3);

ziqi.putShip(b1, p1);
ziqi.putShip(b2, p2);
ziqi.setHit(true,p1);
ziqi.setHit(false, p2);
ziqi.print();

}

```

On peut voir les résultats:

Navires:											Frappes:										
	A	B	C	D	E	F	G	H	I	J		A	B	C	D	E	F	G	H	I	J
0	0
1	B	B	B	B	.	.	1	X
2	2
3	D	D	.	3	X	.	.
4	4
5	5
6	6
7	7
8	8
9	9

Quand il y a un hit, on tape un croix rouge, quand il n'y en a pas, on tape un croix noir, quand il n'y a rien, on tape un point.

Si on appelle `addStrike()` plus d'une fois par `shipState`, le navire pourra donc être touché plus que le permet sa longueur, afin de gérer cet état illégal, on modifie un peu la fonction

`addStrike()` :

```

public abstract class AbstractShip {
    ...
    public void addStrike() {
        if(this.strikeCount < this.taille) this.strikeCount++;
        else this.strikeCount = this.taille;
    }
}

```

6 Exercice 6

On modifie les fonctions `sendHit`, et on ajoute une fonction `exercice6` dans le document `testBoard.java` pour tester:

```

public static void exercice6() {
    Board ziqi = new Board("ziqi");
    AbstractShip b1 = new Battleship();
    AbstractShip b2 = new Destroyer();
    Coords p1 = new Coords(4,1);
    Coords p2 = new Coords(7,3);
}

```

```

        Coords p3 = new Coords(5,1);
        Coords p4 = new Coords(6,1);
        Coords p5 = new Coords(7,7);

        //L'etat initial
        ziqi.putShip(b1, p1);
        ziqi.putShip(b2, p2);
        ziqi.print();
        System.out.println("-----");
    ");

    // il y a un miss
    ziqi.sendHit(p5);
    ziqi.sendHit(p1);
    ziqi.sendHit(p3);
    ziqi.print();
    System.out.println("-----");
    ");

    // il y a un sunk
    ziqi.sendHit(p4);
    ziqi.sendHit(p4);
    ziqi.print();
    System.out.println("-----");
    ");

    //Apres un sunk
    ziqi.sendHit(p5);
}
}

```

A l'état initial, il y a deux bateaux sur la grille et il n'y a pas de hit.

Navires:											Frappes:										
	A	B	C	D	E	F	G	H	I	J		A	B	C	D	E	F	G	H	I	J
0	0
1	B	B	B	B	.	.	1
2	2
3	D	D	.	3
4	4
5	5
6	6
7	7
8	8
9	9

Après, on envoie un hit dans la position **H7**. Comme il n'y a pas de bateau sur cette position, donc on reçu un **MISS**, ensuite on envoie deux hits dans la position **E1** et **F1**, le bateau b1 est couché.

```

-----
Miss
Bien hit
Bien hit
Navires:
  A B C D E F G H I J
0 . . . . . . . . . .
1 . . . . B B B B . .
2 . . . . . . . . . .
3 . . . . . . . D D .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .

Frappes:
  A B C D E F G H I J
0 . . . . . . . . . .
1 . . . . X X . . . .
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . X . .
8 . . . . . . . . . .
9 . . . . . . . . . .
-----

```

Après on envoie deux hits à la position **G1**, comme le bateau n'est pas détruit, il est bat deux fois, après il est détruit, et puis quand on envoie hit à la position **G1** la troisième fois, on reçu un **MISS**.

```

-----
Bien hit
Bien hit
B coulé
Navires:
  A B C D E F G H I J
0 . . . . . . . . . .
1 . . . . . . . . . .
2 . . . . . . . . . .
3 . . . . . . . D D .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . . . .
8 . . . . . . . . . .
9 . . . . . . . . . .

Frappes:
  A B C D E F G H I J
0 . . . . . . . . . .
1 . . . . X X X . . . .
2 . . . . . . . . . .
3 . . . . . . . . . .
4 . . . . . . . . . .
5 . . . . . . . . . .
6 . . . . . . . . . .
7 . . . . . . . X . .
8 . . . . . . . . . .
9 . . . . . . . . . .

Miss
-----

```

7 Exercice 7

On écrit le document `testGame.java`, il faut bien jouer.

```

public class testGame {
    public static void main(String args[]) {
        Board ziqi = new Board("ziqi");
        AbstractShip aShipList[] = {new Destroyer(), new Submarine(), new
Submarine(), new Battleship(), new Carrier()};

        BattleshipsAI ai = new BattleshipsAI(ziqi,ziqi);
        ai.putShips(aShipList);
        ai.printEtat();
        Coords aCoord = new Coords(2,3);
        int times = 1;

```

```

while(ziqi.getDestroy() != aShipList.length) {
    int x = new Random().nextInt(ziqi.getSize());
    int y = new Random().nextInt(ziqi.getSize());
    aCoord = new Coords(x,y);
    System.out.println(times++ +" fois -----
-----");
    ai.sendHit(aCoord);
    ai.printEtat();
}
}
}

```

8 Exercice 8 \ Bonus 1 \ Bonus 2

Il faut bien jouer `Game.java`.

Pour le Bonus 2, je n'ai pas réussi, je ne sais pas la raison.