

# BATAILLE NAVALE

Projet de validation ISN 2016  
de l'académie de Lyon

FRÉDÉRIC MULLER - [maths.muller@gmail.com](mailto:maths.muller@gmail.com)  
LIONEL REBOUL

13 mars 2016



# TABLE DES MATIÈRES

<b>1</b>	<b>Présentation du projet</b>	<b>5</b>
1	Le jeu de la bataille navale . . . . .	5
2	Objectifs du projet . . . . .	5
3	Liste des modules du projet . . . . .	6
<b>2</b>	<b>Noyau du programme</b>	<b>7</b>
1	Le module bn_grille.py . . . . .	7
2	Le module bn_joueur.py . . . . .	8
3	Algorithme de résolution . . . . .	8
4	Autres modules du projet . . . . .	8
<b>3</b>	<b>Affichage console</b>	<b>9</b>
<b>4</b>	<b>Interface graphique</b>	<b>11</b>
<b>5</b>	<b>Guide des modules utilisés</b>	<b>13</b>
<b>6</b>	<b>Point de vue pédagogique</b>	<b>15</b>
<b>7</b>	<b>Conclusion</b>	<b>17</b>

## *TABLE DES MATIÈRES*

# PRÉSENTATION DU PROJET

## 1 Le jeu de la bataille navale

Le jeu de la bataille navale est un jeu qui se joue à deux joueurs.

Chaque joueur dispose d'une grille sur laquelle il place des bateaux rectangulaires de différentes tailles et essaie, chacun son tour, de deviner l'emplacement des bateaux de l'adversaire par des tirs successifs, ce dernier annonçant à chaque coup « manqué » ou « touché ». Nous avons pris le parti de ne pas annoncer « coulé » lorsque toutes les cases d'un bateau ont été touchées pour rendre l'algorithme de résolution un petit peu plus intéressant.

Les bateaux peuvent être placés horizontalement ou verticalement et deux bateaux ne peuvent pas se trouver sur des cases adjacentes.

Les règles retenues dans ce projet sont les règles du jeu original, mais elles peuvent être facilement modifiées, à savoir que la grille est un carré 10 cases de côté et la composition de la flotte est la suivante :

- Un bateau de 5 cases
- Un bateau de 4 cases
- Deux bateaux de 3 cases
- Un bateau de 2 cases

Notons tout de suite quelques implications stratégiques de ces règles qui seront utilisées dans l'algorithme de résolution :

- Le plus petit bateau étant de taille 2, il suffit de ne tirer que sur une cases sur 2 (imaginez les cases noires d'un damier) lors de la recherche d'un bateau.
- Une fois qu'un bateau a été coulé (soit parce que c'est le plus grand de la liste, soit parce que les cases adjacentes à ses extrémités ont été manquées), on peut éliminer de la recherche toutes ses cases adjacentes.

## 2 Objectifs du projet

Nos objectifs ont été les suivants :

- Définir une structure de données pour modéliser la grille de jeu, ainsi que les joueurs.
- Implémenter un algorithme de résolution par l'ordinateur qui soit le plus performant possible (en nombre de coups ainsi qu'en temps de résolution d'un grille) et en faire une étude statistique complète.
- Avoir une interface permettant de jouer contre l'ordinateur. Cette interface a été réalisée d'un part en mode console avec un affichage grâce à des caractères graphiques (en unicode) et, d'autre part, avec le module tkinter.

### 3 Liste des modules du projet

Afin de faciliter le développement et la maintenance du projet, celui-ci a été décomposé en un certain nombre de modules :

- `main.py` : le programme principal. Il permet, via un argument `-interface` en ligne de commande de choisir l'interface de jeu (`console` ou `tkinter`).
- `bn_utiles.py` : contient quelques fonctions utiles ainsi que les constantes du projet.
- `bn_grille.py` : gère la grille et les bateaux.
- `bn_joueur.py` : gère les joueurs et implémente l'algorithme de résolution.
- `bn_console.py` : toute l'interface en mode console, et l'étude statistique de l'algorithme de résolution.

# NOYAU DU PROGRAMME

Dans ce chapitre nous allons détailler les structure de données utilisées ainsi que l'algorithme de résolution.

## 1 Le module `bn_grille.py`

Ce module définit la grille de jeu. Il permet de gérer les bateaux sur la grille, ainsi que l'état de chaque case.

### 1.1 La classe `Bateau`

Cette classe, très minimaliste, définit un bateau par sa case de départ, sa taille et sa direction. Elle permet de récupérer :

- sa case de fin,
- la liste de ses cases occupées,
- la liste de ses cases adjacentes.

### 1.2 La classe `Grille`

Cette classe est l'une des principales du projet. Elle gère l'ensemble de la grille et permet de mémoriser :

- les bateaux du joueur,
- les cases jouées et leur état (touchée ou manquée)

L'attribut `Grille.etat` fournit l'état de la grille. C'est un dictionnaire indexé par les couples  $(0,0)$ ,  $(0,1)$ , ...,  $(9,9)$ , dans lesquels la première coordonnée correspond à la colonne de la case et la deuxième à sa ligne.

L'état d'une case peut être :

- 0 : case non jouée
- 1 : case touchée
- -1 : case manquée ou impossible

L'intérêt d'utiliser un dictionnaire plutôt qu'une double liste tient au fait que les appels sont plus simples et plus naturels et, surtout, que l'utilisation d'une table de hachage permet la recherche d'un élément en  $O(1)$ .

- 2 Le module bn\_joueur.py**
- 3 Algorithme de résolution**
- 4 Autres modules du projet**



## **AFFICHAGE CONSOLE**



# INTERFACE GRAPHIQUE



## **GUIDE DES MODULES UTILISÉS**



## **POINT DE VUE PÉDAGOGIQUE**





## CONCLUSION