

# Projet bataille Naval

Introduction : Création d'un jeux de bataille Naval en python et réalisation d'un rapport

Membre : Jules Loiseau 107, Eloi Leclair 107

Contexte : projet de cours de première NSI

Objectif : Un jeux de bataille Naval fonctionnel avec un affichage à chaque manche joué ainsi qu'un rapport détaillant tout le projet.

Cahier des charge :

Fonctionnalité	Contrainte
Utilisation d'un plateau de jeu	Taille 10x10 (plateau bot et joueur)
P-uplet, liste ou dictionnaire	utilisation d'aux moins deux type construit
Placement des bateaux sur le plateau	6 bateaux: 2 de longueur 2, 3 de longueur 3 et 1 de longueur 5
Le bot place aléatoirement c'est bateaux	Taille du plateau et des bateaux à respecter
chaque tour le joueur tire sur une case adverse	Taille du plateau et reconnaissance de la case sur laquelle le joueur tir
Chaque tour le bot tire sur une case aléatoire adverse	Taille du plateau reconnaissance de la case sur laquelle le bot tir
chaque tour mise a jour visuel	interface visuel dynamique dans le terminal sous forme de matrice
Affichage de fin de partie	comptage et vérification des bateaux restant
Commentaire et Docstring	Rendu du code propre et lisible par n'importe qui

Méthodologie et répartition du travaille par fonction:

Eloi :

- create\_board\_player
- add\_boat
- select
- create\_board\_bot

Jules :

- create\_board\_player
- compare
- win
- refresh
- pick\_up\_infos

## Fonction :

- création du plateau du joueur
- ajout des bateau pour le joueur
- création du plateau bot avec placement des bateaux aléatoirement
- selection de la case sur laquelle le joueur veut tiré ou aléatoirement pour le bot
- comparaison de la case indiqué par le joueur ou bot dans la fonction select avec celle du plateau pour vérifier si un bateaux si trouve
- rafraichissement du plateau après chaque tir de bot ou de joueur
- condition permettant la fin du jeux lorsque tout les bateaux du joueur ou du bot sont coulé

## Fonction optionnel :

- taille du plateau et nombre de bateaux de chaque taille réglable par le joueur (dans les limites du possible)
- interface graphique pour le plateau de jeux
- plusieurs difficulté de bot
- joueurs 1vs1

## Date de rendu: 19 Janvier au soir

## Docstring des fonction :

create\_board\_player:

parametres --> taille de tableau (int) \ nombre de case avec des bateaux (int)

retourne --> tableau (matrice)

rôle --> creer une tableau et permet au joueur de placer ses bateaux grâce a la fonction

add\_boat()

qui --> fait à 2

add\_boat:

parametres --> taille de tableau (int) \ nombre de case avec des bateaux (int) \ tableau (matrice)

\ dictionnaire de coordonnées (dictionnaire)

retourne --> tableau (matrice)

rôle --> intervient dans la fonction create\_board\_player et permet au joueur de placer ses bateaux

qui --> Eloï

select:

parametres -> qui: bot/joueur (str)

retourne --> position (tuple)

rôle --> permet de récupérer des coordonnées données par le joueur ou aléatoirement par le bot

qui --> Eloï

create\_board\_bot:

parametres --> taille de tableau (int) \ nombre de case avec des bateaux (int)

retourne --> tableau (matrice)

rôle --> creer une tableau et place aléatoirement les bateaux

qui --> Eloï

compare:

parametres --> tableau (matrice) \ coordonnées (tuple)  
retourne --> booléen  
rôle --> vérifier si il y à un bateau au coordonnées rentrées  
qui --> Jules

win:

parametres --> tableau (matrice)  
retourne --> booléen  
rôle --> vérifier si un utilisateur a gagné  
qui --> Jules

refresh:

parametres --> tableau (matrice)  
retourne --> /  
rôle --> rafraichir l'affichage du jeu à chaque changement sur le plateau  
qui --> Jules

optionnel:

pick\_up\_infos:

parametres --> /  
retourne --> taille de tableau (int) \ nombre de cases avec des bateaux (int)  
rôle --> récupérer les parametres du jeu données par le joueur  
qui --> Jules