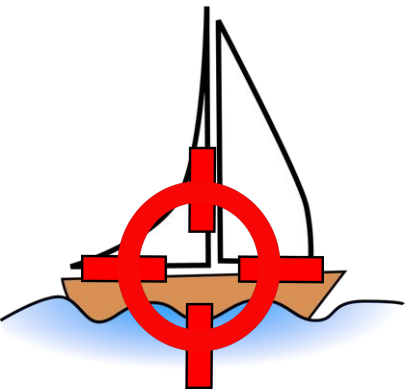


Touché-coulé

La bataille navale avec l'IA



Benjamin MAURICE, Mehdi MUNIM



La bataille navale

La bataille navale est un jeu de société à jouer à deux.

Deux opposants placent des bateaux sur une grille sans connaître la position de l'ennemi. Le but du jeu est de couler tous les navires adverses.

Explications en image dans les slides suivantes...



Voici à quoi ressemble
une grille



	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Le joueur 1 commence
par placer ses navires sur
la grille

Le Joueur 1

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

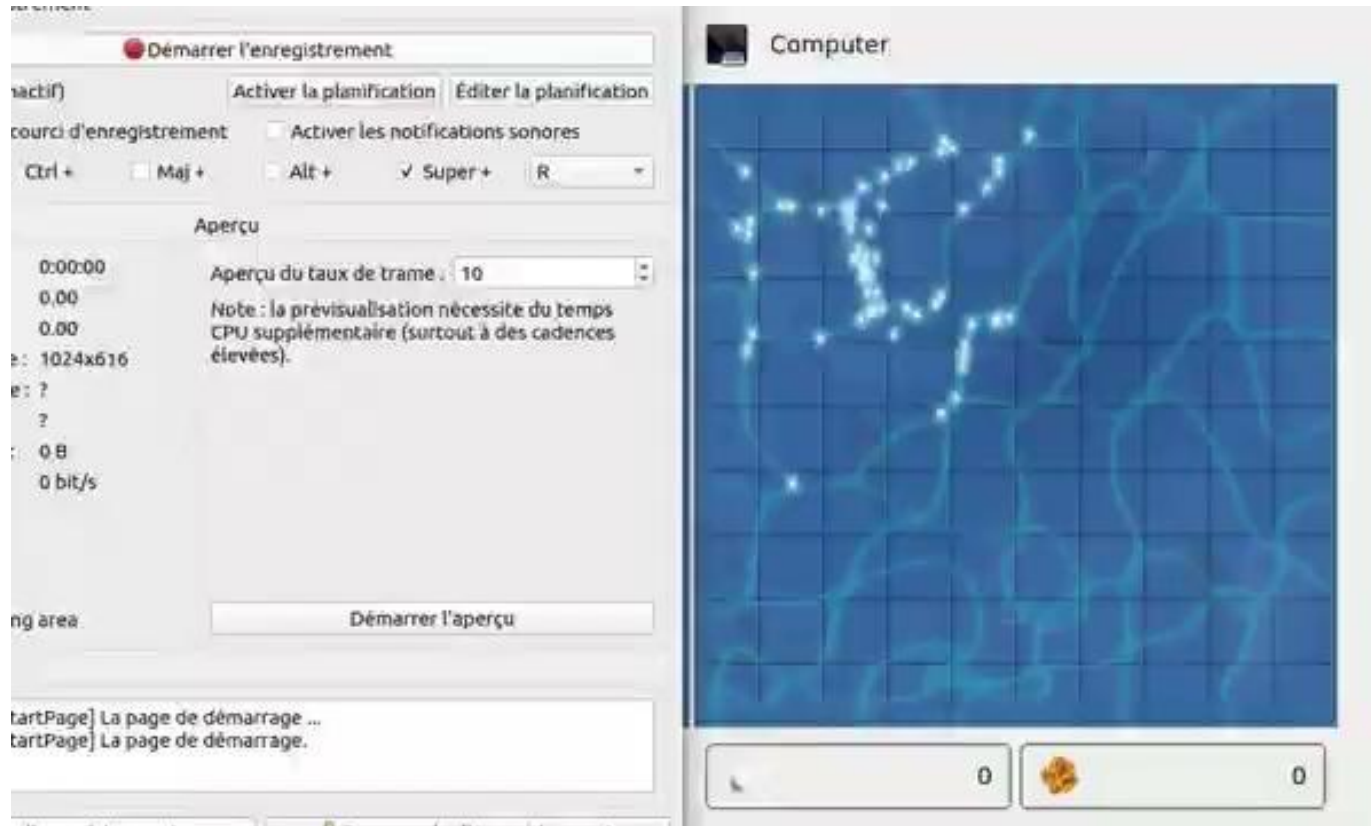
Le joueur 2 ne voit pas les positions adverses.

Il choisit une case, et essaye de toucher les navires jusqu'à tous les couler.

Le Joueur 2

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Démo KNavalBattle (Linux)



Objectifs de la séance

Dans cette activité, nous allons nous servir de Python pour élaborer trois stratégies (donc trois IA) pour jouer à la bataille navale.

- Stratégie 1 : Aléatoire
- Stratégie 2 : Hunt / Target
- Stratégie 3 : Probabilités

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4			X							
5						X	X			
6		X						X		X
7				X						X
8	X	X						X		
9										
10										

Prérequis

- Avoir téléchargé [le code template](#) sur ton ordinateur
- Avoir Python installé sur l'ordinateur, avec si possible un éditeur de code (Pycharm, VSCode...)

Commençons !

Télécharge le code source

Clique sur ce lien :

<https://drive.google.com/file/d/1CIIQQ7hneogHQP9mS8zB2brSoBHVg30T/view?usp=sharing>

Télécharge le code source. Il devrait apparaître dans ton dossier “Téléchargements”..

Vérifie si Python 3 est installé

- Ouvre l'**invite de commande (cmd)**. Pour ce faire, appuie sur la touche Windows, tape "cmd" dans la barre de recherche, puis appuie sur Entrée.
- Dans l'invite de commande, tape la commande suivante :

python3 --version

Si Python 3 est installé, tu verras la version installée. Sinon, un message d'erreur indique que la commande n'est pas reconnue



Installe les bibliothèques Python nécessaires

Dans l'**invite de commande (cmd)**, télécharge les bibliothèques :

- seaborn
- matplotlib
- pygame
- numpy

Avec la commande :

pip install <nom de la bibliothèque>



Commençons doucement...

Complète les lignes commençant par `###` dans les fonctions `place_ships`, `draw_board`, `shoot` et `play`

```
def place_ships(self):  
    """  
  
    Cette fonction doit placer aléato  
  
    Complète start_row et start_col  
  
    en choisissant un nombre entre 0
```

```
def shoot(self, guess_row, guess_col):  
    self.SHOT_MAP[guess_row][guess_col] = 1  
  
    ### Ajoute 1 à la variable self.NUM_GUES  
  
    if self.SHIP_MAP[guess_row][guess_col] =  
        self.SCORE += 1  
    ship = self.COORDINATE_SHIP_DICT.pop
```

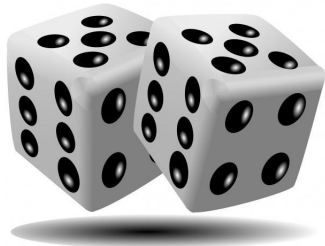
```
def play(self):  
    self.place_ships()  
    while True:  
        for event in pg.event.get():  
            if event.type == pg.QUIT:  
                pg.quit()  
                sys.exit()  
            if event.type == self.GUESS_EVENT:  
                guess_row, guess_col = self.g
```

```
def draw_board(self):  
    board_y = self.BOARD_Y  
  
    ### Itère une variable i de 0 à 9  
  
    board_x = self.BOARD_X  
    for j in range(10):  
        if self.SHIP_MAP[i][j] ==  
            pg.draw.rect(SCREEN, V
```

Stratégie 1 : Aléatoire

Avec cette méthode, l'**IA tire sur une case au hasard** (toutes les cases ont autant de chance d'être touchées). Ce n'est pas la meilleure idée mais elle nous servira à comparer de meilleures méthodes

Complète la fonction `guess_random`



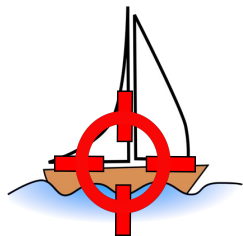
```
def guess_random(self):  
    """  
    Choisit au hasard  
    un numéro de case (colonne, ligne entre 0 et 9)  
  
    Attention cependant à choisir une case où tu n'as pas  
    encore tiré  
    donc où self.SHOT_MAP[ligne][colonne] est différent de zéro.  
  
    """  
    ligne_hasard = 0  
    col_hasard = 0  
  
    return ligne_hasard, col_hasard
```

Stratégie 2 : Hunt / Target

Dans cette méthode, l'IA choisit une case au hasard **sauf si elle touche un bateau.**

C'est logique : tu as touché une case d'un bateau, une case à côté appartient sûrement au navire.

Implémente la fonction **hunt_target** en t'aidant des lignes marquées **##**



```
def hunt_target(self, length=None):  
    """  
    Méthode Hunt / Target  
  
    Complète les lignes commençant par ##  
    """  
    # enter hunt mode when no more targets left  
    if not self.targets:  
        ## choisit une ligne et une colonne au hasard  
        ## avec la fonction self.guess_random()  
        ## nomme les guess_row (ligne) et guess_col (colonne)  
    else:  
        guess_row, guess_col = self.targets.pop()  
  
    # Si la case a déjà été touchée  
    if self.GUET.MAP[guess_row][guess_col] == 1:
```

Stratégie 3 : Probabilités

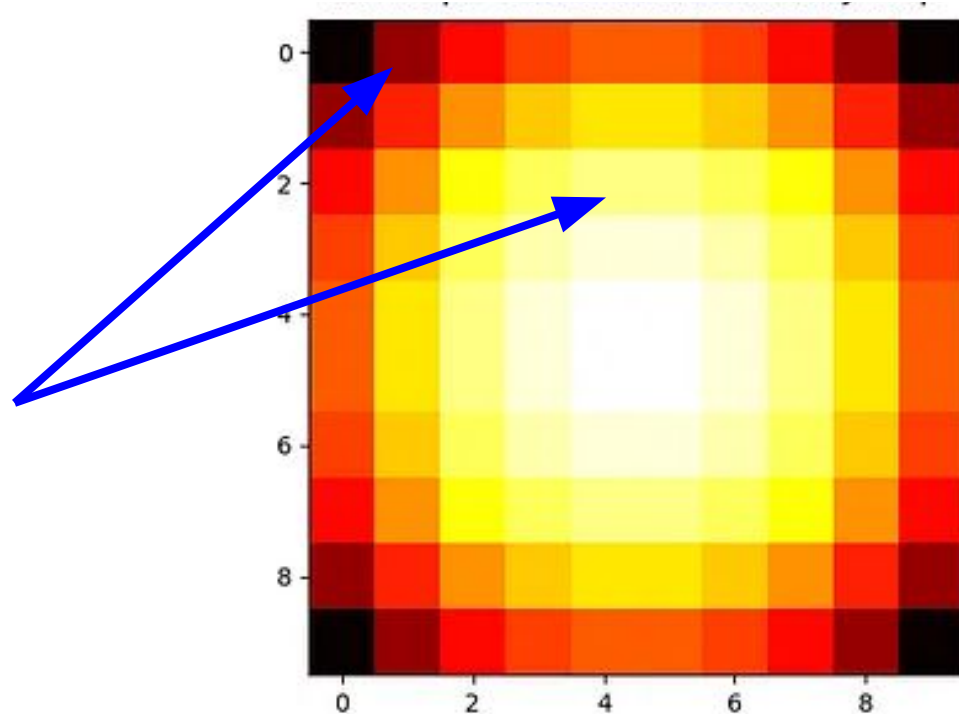
En réalité, les cases du centre ont beaucoup plus de chance d'être occupées par un navire. On a intérêt à les viser !

Complète les fonctions `gen_prob_map` et `guess_prob` qui s'en occupent

$$\begin{array}{l} 2 > -3 \\ 0.999... = 1 \\ \pi \approx 3.14 \\ \sqrt{2} \\ 1 + 2 \cdot 3 \\ 5(2 + 2) \\ 101_2 = 5_{10} \\ \infty \\ + - \\ \times \div \\ 5^2 \\ (1 - 2) + 3 \end{array}$$

Stratégie 3 : Probabilités

Les navires ont plus de chance d'être sur une case blanche que rouge



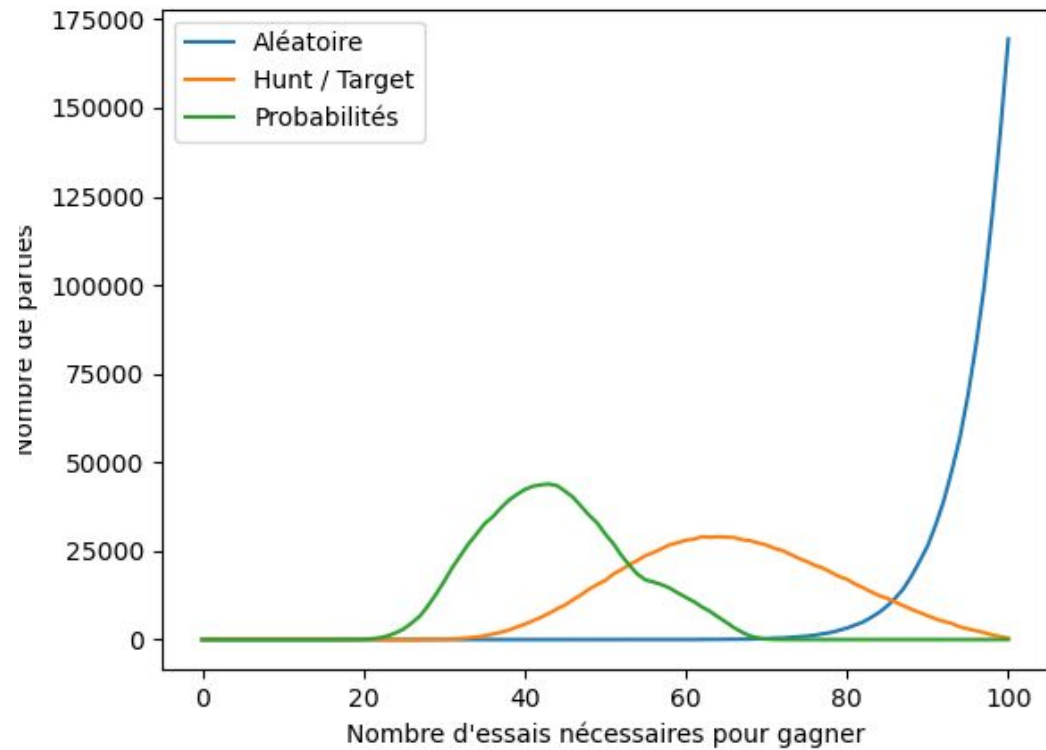
Comparaison

Nous avons déjà testé les 3 méthodes pour toi, et obtenu la figure de la slide d'après.

On représente le nombre de coups qu'il faut pour gagner.

Par exemple, en mode "aléatoire", il faut presque 100 parties pour que gagner, mais en mode "hunt / target", 70 coups suffisent.

Simulation sur un million de parties



Comparaison

Qu'en pense-tu ? Quelle est la meilleure méthode d'après la figure ?

Expérimente par toi-même. Avec ton groupe, note combien de coups te sont nécessaires pour gagner.

Quelle méthode obtient les meilleurs résultats ?

Source (Article)

Coding an Intelligent Battleship Agent

Developing a strategy from random guessing to superhuman performance



Aydin Schwartz · Follow

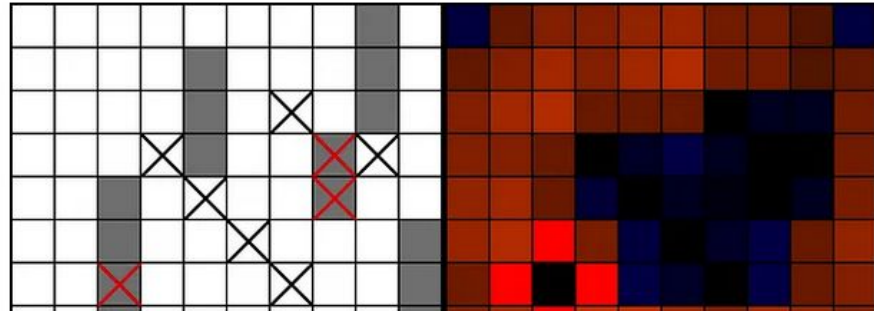
Published in Towards Data Science · 8 min read · Apr 15, 2022



127



1



Annexe : installer Python 3

Si Python 3 n'est pas installé, vous pouvez le télécharger et l'installer à partir du site officiel de Python. Voici les étapes à suivre :

- Ouvrez votre navigateur web et accédez au site officiel de Python à l'adresse <https://www.python.org/>.
- Cliquez sur le bouton "Downloads" (Téléchargements) dans le menu principal.
- Téléchargez la dernière version de Python 3 en cliquant sur le lien de téléchargement approprié pour Windows.
- Une fois le téléchargement terminé, exécutez le programme d'installation.
- Suivez les instructions à l'écran pour installer Python 3. Assurez-vous de cocher la case "Add Python 3.x to PATH" (Ajouter Python 3.x au PATH) pendant l'installation. Cela permettra d'accéder à Python depuis n'importe quel répertoire dans l'invite de commande.

En dernier recours, lève la main
puis attends tranquillement ton
formateur :)