SORBONNE UNIVERSITÉ

3I005
Projet 1 :
Bataille navale
Probas, Stats et Info

Réalisé par : **BOUSBA Abdellah**
Encadré par : **Tom Veniat**

3ème année Licence Informatique
Mono-disciplinaire
Groupe n°

Sommaire

1. Modélisation et fonctions simples : 3

2. Combinatoire du jeu :...3

3. Modélisation probabiliste du jeu : 4

3.1. Version aléatoire : 4

3.2. Version heuristique : 7

3.3. Version probabiliste simplifiée : 7

1. Modélisation et fonctions simples :

Dans cette partie on définie nos objets et fonction de base, le choix des classes était le suivant :

Class Bateau : représente un bateau et redéfinie le mécanisme de comparaison pour cette objet.

Class Grille : représente notre grille et contient la matrice et la liste des bateaux. ***Pour réduire la complexité, la comparaison entre deux grilles est basée sur l'égalité des listes des bateaux***.
Le placement aléatoire est basé sur une méthode ***sans remise*** pour optimiser le temps de génération.

Le reste des fonctions est suffisamment documenter et facile a comprendre.

2. Combinatoire du jeu :

2.1. nombre de configurations possibles :

En prenant le nombre maximum de nombre configuration possible pour l'emplacement de chaque bateau de la liste [2, 3, 3, 4,5] sur une grille de taille 10 on aura :

nb = 180 × 160 × 160 × 140 × 120 = **77414400000** avec le nombre de configuration pour placer un bateau de longueur l sur une grille vide de taille N est (N-l+1) × N × 2

Par contre pour aller plus loin on peut placer chaque bateau le plus a gauche et le plus a en haut possible et faire le calcule a la main on aura :

9 × 10 × 2
+ 7 × 2 + 8 × 8 + 8 × 9 + 6
+ 7 × 5 + 8 × 5 + 8 × 9 + 3
+ 6 × 8 + 7 × 2 + 7 × 9
+ 4 × 4 + 4 × 5 + 2 × 6 + 6 × 9 = 180 × 156 × 150 × 125 × 102 = **53703000000**.

2.2. en utilisant la fonction denombre_bateau_vide on obtient exactement le même résultat.

2.3. Nombre de configuration exacte :

Pour une liste [2] : 180
Pour une liste [2,3] : 27336
Pour une liste [2, 3, 3] : 3848040

En essayant d'exécuter la fonction pour une liste de 5 ou 4 bateaux, 8 heures d'exécution n'est pas suffisante pour avoir le résultat.

2.4. Soit nb : le nombre de grilles et p : la probabilité de tirer une grille donnée, comme toutes les grilles sont équiprobables alors p = 1 / nb.

2.5. En se basant sur la question précédente il suffit de calculer la probabilité p d'avoir une grille donnée et appliqué la formule nb = 1 / p, et pour cela on calcule le nombre de grille génère pour avoir la même grille pour un nombre d'itération (dans notre cas 10 itération) et diviser le résultat sur le nombre d'itération pour avoir une approximation de la probabilité d'où le nombre de grille possible.

Pour une liste [2] : 169
Pour une liste [2,3] : 29214
Pour une liste [2, 3, 3] : 2405116

Comme on remarque ci-dessus l'approximation pour une liste de 1 ou 2 bateaux est plus ou moins proche du résultat exacte par contre pour une liste de 3 bateaux on est loin de plus de 1 million, et ca peut être expliqué par le fait que le nombre de grille générées a chaque fois peut être très inferieur ou très supérieur au nombre exacte d'où la moyenne sera loin du résultat attendu, la solution est de tirer aléatoirement pour un grand nombre d'itération mais ca sera lourd en coté de temps d'exécution :
Prenant 100 comme nombre d'itération et une liste de 5 bateaux.
Supposant que la comparaison entre deux grilles se fait en 0.01 ms, et le nombre de configuration attendu est la borne supérieure (cf. question 2.1) 53703000000.
L'exécution sera terminer dans 53703000000 × 100 × 0.01 ms = 1.7 ans
Alors cette méthode n'est pas une bonne méthode pour calculer le nombre de configuration

3. Modélisation probabiliste du jeu :

Dans cette partie on introduit le concept du joueur :

Class Bataille : représente les caractéristique d'un jeu et les fonctions nécessaires pour jouer.
Class Joueur : contient le type du joueur et la version du jeu qui correspond

3.1. Version aléatoire :

On suppose que :

- Les tires sur les cases sont équiprobables (on a la même probabilité de tirer sur chaque case).
- On ne peut pas tirer sur la même case deux fois (sans remise des cases déjà touché)
- Ne pas gagner si on tire k fois est l'événement complémentaire de gagner si on tire k fois.
- La grille est de taille 10 et contient 5 bateau de taille : [2, 3, 3, 4, 5]

En se basant sur les hypothèses ci-dessus on calcule la probabilité P_{win_k} de gagner si on tire k fois :

Soit $(P_{touché})_i$ = nombre de cases de bateau restantes/nombre de cases restantes dans la grille la probabilité de toucher un bateau au i-eme tire

et $(P_{nonTouché})_i = 1 - P_{touché}$ = nombre de cases vides restantes/nombre de cases restantes dans la grille la probabilité de ne pas toucher un bateau au i-eme tire

$$P_{win\_k} = ((P_{nonTouché})_1 \cap (P_{nonTouché})_2 \cap \ldots\ldots\ldots \cap (P_{touché})_{k-1} \cap (P_{touché})_k) \cup$$

$$((P_{nonTouché})_1 \cap (P_{nonTouché})_2 \cap \ldots\ldots\ldots \cap (P_{nonTouché})_{k-1} \cap (P_{touché})_k) \cup \ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots\ldots$$

$$\ldots\ldots\ldots \cup ((P_{touché})_1 \cap (P_{nonTouché})_2 \cap \ldots\ldots\ldots \cap (P_{nonTouché})_{k-1} \cap (P_{touché})_k) \cup$$

$$((P_{touché})_1 \cap (P_{touché})_2 \cap \ldots\ldots\ldots \cap (P_{nonTouché})_{k-1} \cap (P_{touché})_k)$$

En utilisant la formule des probabilités conditionnelles on calcule P_{win_k} pour chaque k=1,…,100

Soit X la variable aléatoire qui représente le gain en r = 1,…,100 fois :

$$P(X = r) = (P_{nonTouché})_1 \cap (P_{nonTouché})_2 \cap \ldots\ldots\ldots \cap (P_{touché})_r$$

$$P(X = r) = (P_{nonTouché})_1 \times (P_{nonTouché}) \times \ldots\ldots\ldots \times (P_{touché})_r$$

On a: l'Esperance de X $\quad E(X) = \sum_{r=1}^{100} (x_i \times P(X = x_i))$

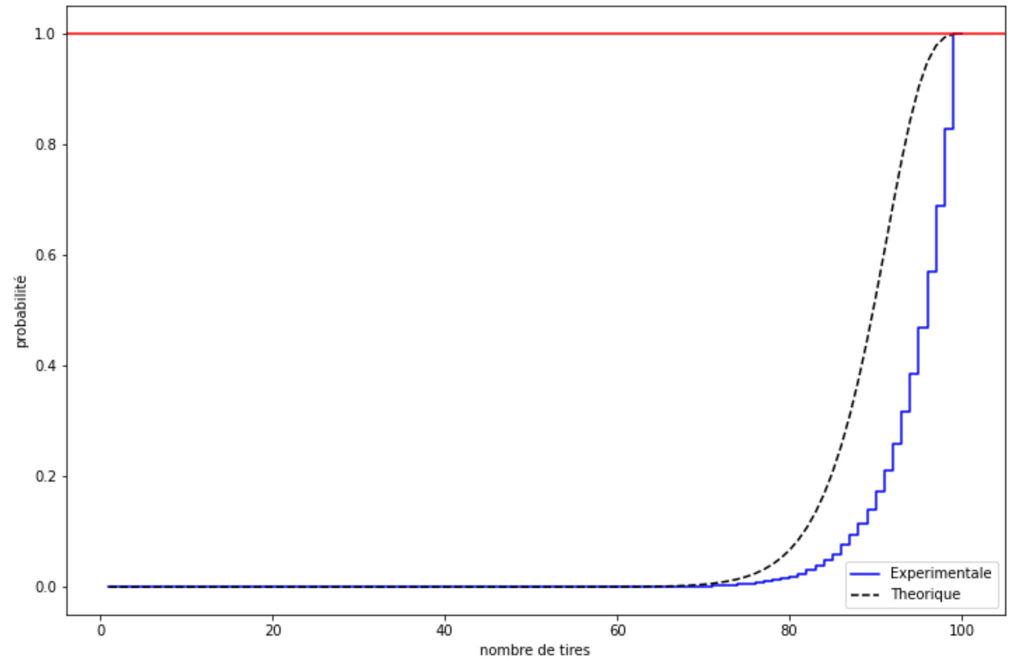
On faisant les calcule on trouve **E(X) = 89.37** (ci-dessous le tableau de probabilité de pour chaque r).

```
[0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 1.50372890e-19 2.57137642e-17 1.71425095e-16
 8.99981747e-16 3.95991969e-15 1.51796921e-14 5.20446587e-14
 1.62639559e-13 4.69847614e-13 1.26858856e-12 3.22913451e-12
 7.80374173e-12 1.80086348e-11 3.98762627e-11 8.50693604e-11
 1.75455556e-10 3.50911111e-10 6.82327161e-10 1.29283041e-09
 2.39173625e-09 4.32790369e-09 7.67219287e-09 1.33429440e-08
 2.27941957e-08 3.82942479e-08 6.33327922e-08 1.03208988e-07
 1.65871571e-07 2.63106586e-07 4.12200210e-07 6.38245224e-07
 9.77312375e-07 1.48077488e-06 2.22115903e-06 3.30000037e-06
 4.85831785e-06 7.09048349e-06 1.02624691e-05 1.47357019e-05
 2.09980657e-05 2.97039570e-05 4.17257477e-05 5.82195439e-05
 8.07087589e-05 1.11189758e-04 1.52264688e-04 2.07307567e-04
 2.80670768e-04 3.77940155e-04 5.06248207e-04 6.74655411e-04
 8.94610714e-04 1.18050154e-03 1.55030220e-03 2.02632546e-03
 2.63607405e-03 3.41317526e-03 4.39835871e-03 5.64040145e-03
 7.19690956e-03 9.13472391e-03 1.15296226e-02 1.44648334e-02
 1.80276652e-02 2.23033176e-02 2.73646706e-02 3.32566462e-02
 3.99737359e-02 4.74297373e-02 5.54200731e-02 6.35798197e-02
 7.13453980e-02 7.79350875e-02 8.23724066e-02 8.35838138e-02
 8.06008632e-02 7.28746940e-02 6.06537429e-02 4.52813981e-02
 2.91739479e-02 1.52405228e-02 5.75050944e-03 1.17781519e-03]
```

On peut calculer l'Esperance expérimentalement en calculant les probabilités P (X = r) de la même manière qu'avant (cf. 2.5) mais en utilisant 10000000 partie comme la complexité de jouer n'est pas trop chère en termes de temps d'exécution. Ensuite calculer E(X) avec ces valeurs.

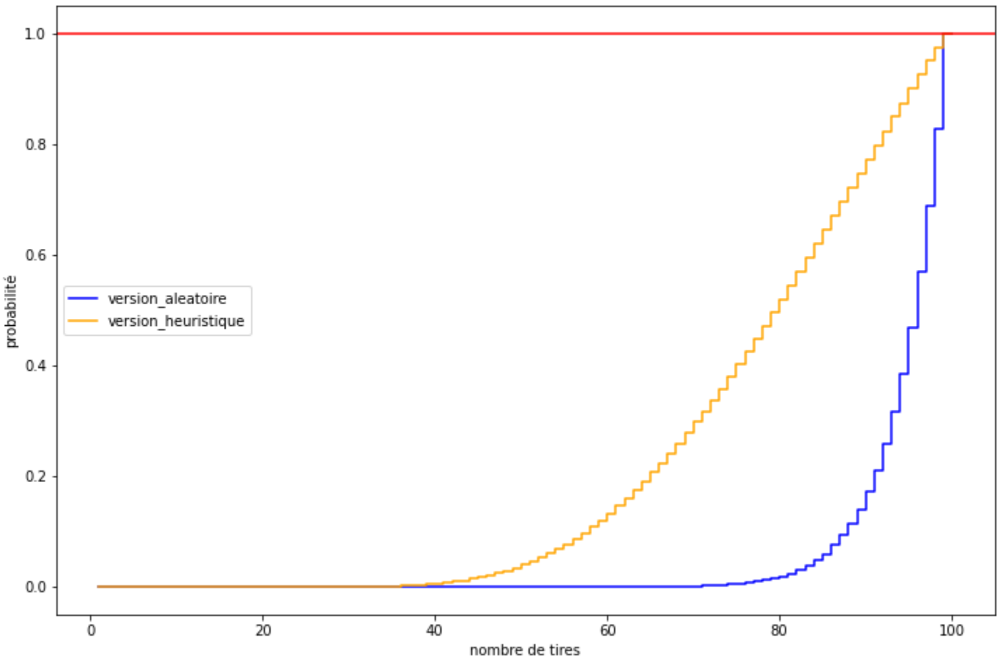
```
[0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 1.0000e-05 1.0000e-05 1.0000e-05
 0.0000e+00 0.0000e+00 1.0000e-05 1.0000e-05 2.0000e-05
 3.0000e-05 1.0000e-05 5.0000e-05 6.0000e-05 6.0000e-05
 9.0000e-05 7.0000e-05 2.1000e-04 1.8000e-04 2.3000e-04
 3.3000e-04 5.8000e-04 5.2000e-04 8.9000e-04 1.1000e-03
 1.3500e-03 1.8200e-03 1.9700e-03 2.7500e-03 3.1100e-03
 3.9100e-03 4.6600e-03 6.2700e-03 7.6900e-03 9.4000e-03
 1.1800e-02 1.4580e-02 1.8340e-02 2.2020e-02 2.6500e-02
 3.2350e-02 3.8990e-02 4.6620e-02 5.8780e-02 6.8620e-02
 8.2380e-02 1.0017e-01 1.1901e-01 1.4259e-01 1.6984e-01]
```

Le résultat est **E(X)=95.38**. En comparant avec le résultat théorique on trouve que le résultat expérimental est légèrement plus grand et ca peut être lié a notre hypothèse que les tires sont équiprobable car expérimentalement tout dépend de la fonction de Radom qu'on utilise en plus notre fonction ne pouvait pas calculer les probabilités trop petit inferieur a 10^{-6} comme la plus petite probabilité théorique est inferieure a 10^{-19} on pourra avoir une meilleur estimation de l'Esperance si on utiliser 10^{19} parties.



3.2. Version heuristique :

En calculant l'Esperance de la même façon avec la version heuristique du jeu, on obtient **E(X)=78.65**. Qui veut dire qu'avec cette version on peut minimiser le nombre de tire pour gagner mieux que la version aléatoire.



3.3. Version probabiliste simplifiée :

Il est déconseiller de calculer la distribution jointe sur tous les bateau car ca sera difficile d'énumérer toutes les grilles ou un bateau se trouve dans une case donnée en terme de temps (comme vu précédemment).

Dans la méthode proposée pour chaque case on obtient le nombre de fois où le bateau $\mathbf{b_x}$ apparaît potentiellement, en divisant par le nombre de configurations possible pour ce bateau dans une grille vide on obtient $P(b_x)$ la probabilité que bi soit dans une la case (i,j)

la probabilité jointe de la présence d'un bateau b_x sur une case (i,j) avec une liste de **k** bateau

$P(b_x \cap (\bigcap_{h=1}^{k} \overline{b}_h)) = P(b_x) \times P(\overline{b}_{h=1}) \times \ldots \times P(\overline{b}_{h=k})$ avec $h \neq x$ (hypothèse : positions indépendantes)

Par contre cette hypothèse est fausse, <u>justification :</u>

Soit une grille de taille 10 avec deux bateau b_1, b_2 de taille 2 pour la case (0,0). Alors :

$P(b_1) = 2/180$ et $P(b_2) = 2/180$. $P(b_1) \times P(b_2) = 0.00012$ $\neq P(b_1 \cap b_2) = 0$ alors la préséance (ou absence) d'un bateau sur une case connexe influence la probabilité.

On propose la solution suivante pour résoudre le problème :

- pour la liste des bateau restants on prend le bateau de taille min.
- on calcule pour chaque case la probabilité que ce bateau soit dans la case.
- à chaque tour de boucle on prend la case de probabilité max
- au fur et à mesure des tires :
    - ➢ si on tire sur une case vide, on prend en compte l'impossibilité d'avoir un bateau dans cette case, et on recalcule sa probabilité de celle des cases connexes.
    - ➢ si on tire sur un bateau et ce n'est pas sa dernière case, on explore les cases connexes (haut, bas, droite, gauche) par ordre décroissant de probabilité.
    - ➢ si on tire sur un bateau et c'est dernière case on est sur que c'est la position de ce bateau alors on le place sur la grille et recalcule la probabilité

En calculant l'Esperance de la même façon avec la version heuristique du jeu, on obtient **E(X)=56.88**.
Qui est la meilleure version pour gagner en minimisant les tires.