27/12/2021.

* MENACE = Toujours bot qui commence
* Compréhension du sujet
* Structure possible : liste chainée, table hachage
* 2 Idées :  
  - enregistrer toutes les grilles possible (en passant par des fonctions qui enlever les cas impossible et les répétitions) et on compare un par un pour trouver la boite correspondant, utilisation de table hachage pour stocker tous les similaire de chaque.  
    
  - Ignorer tous les similaires, on garde juste la grille initiale et pour trouver la boite, on fait la comparaison de tous les rotations avec les 304 possibilité (sans les similaires) donc dans le pire des cas, on aura 304\*8 (2432 fois )comparaison pour la trouver. (Donc peut être plus efficace que d’enregistrer toutes les boites) (mais cela nécessite une fonction qui permet d’enlever toutes les grilles similaires.)
* Etape nécessaire pour réaliser ces 2 idées  
  - Traiter les 15000+ grilles en enlevant les cas impossible + les répétions
* Commencer par coder les fonctions de base :  
  - Transformer les grilles en base 3 puis en base 10 et inverse pour reproduire la grille  
  - Coder le jeu de morpion (les règles + menu) et sous forme d’un tableau 2d

28/12/2021

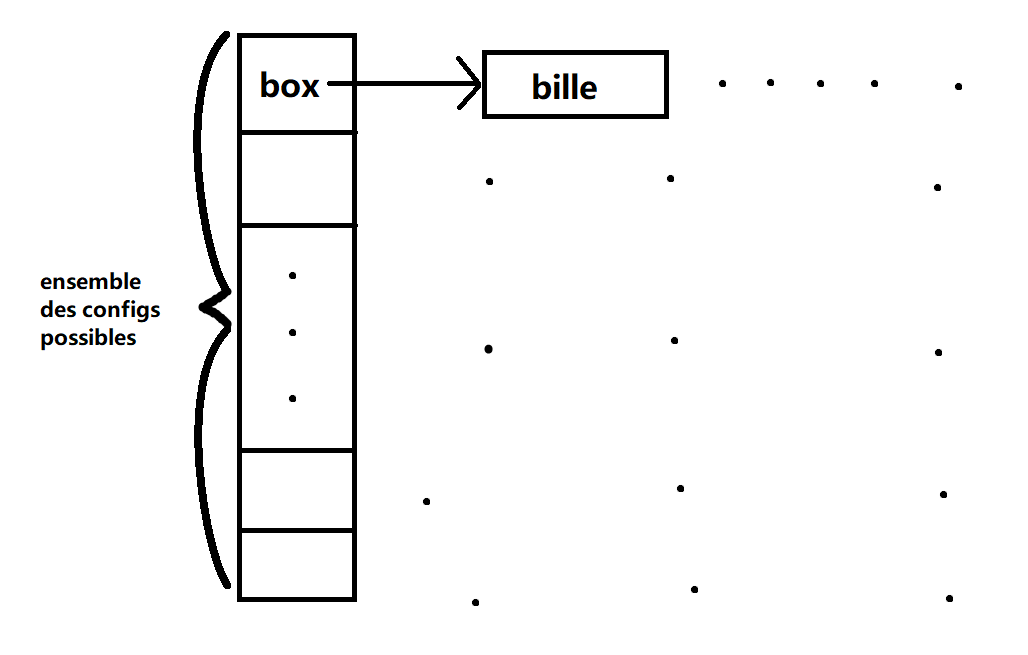
* Codage du jeu de base avec 2 jouer
* Il est mieux de manipuler un tableau 1d que 2d (pour mieux répondre quand robot tire au sort un bille entre 9, donc il est mieux de manipuler un tableau 1d)
* Finir le jeu de base (joueur vs joueur)

29/12/2021

* Terminer le grille to base 10 et base 10 to grille
* J’ai repris le code d’aide projet et changer les tab 2d en 1d car mon idee est de traiter de table 1d qui sera plus facile pour la suite
* J’ai fait des fonctions qui liste tous les combinaisons possibles de la grille en éliminant certains cas comme : les cas impossibles, les cas déjà gagne et les similaire (en transformant tous les similaires de chaque cas en base 10 puis comparer avec tous les autres cas ce qui fait normalement 6617\*7 comparaison à faire)

30/12/2021

* Utilisation de tableau d’hachage :



* L’idées :
  + 1er étape : affecter les billes dans chaque Box en fonction de sa configuration avec fonction init\_tab\_box() (tous les configurations (en base 3) seront stockées dans un fichier auparavant : b3Config.txt, pour pouvoir lire les indices des 0 et ajouter les billes correspond à ce positon)
  + 2e: On tire au sort entre 0 et taille de box (qui est déjà défini après étape 1), et en fonction du numéro qu’on a tiré, on cherche dans box la bille pour laquelle a comme indice le numéro qu’on a tiré au sort et enregistrer dans un tableau.

Exemple : indice\_bille\_joue = [2, 3, 4 ……] pour pouvoir faire les modifications plus tard.

* + 3e : On enregistre les indices des configurations jouées durant le jeu.

Exemple : indice\_config\_joue = [3,12,31 ……].

* + 4e :

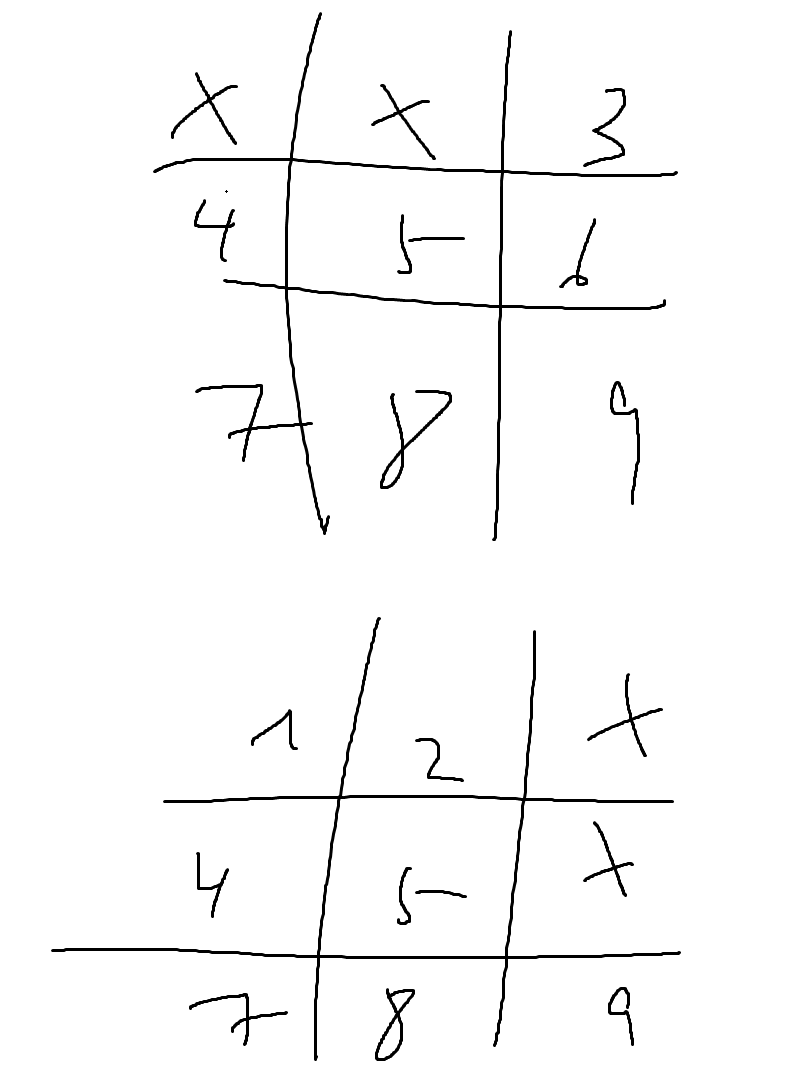
**S’il gagne** **ou nul** : en fonction des tableaux indice\_config\_joue et indice\_bille\_joue, on trouve la position de la bille avec la fonction data\_bille\_tabbox() et ajouter dans la queue de la liste 3 billes(maillons) ou 1 bille de la même position.

**S’il perte** : item pour le début et on retire la bille avec la fonction rem\_position\_tabbox pour l’enlever du box.

* J’ai réussi a trouvé les 304 combinaisons sur internet et le mettre en base 10 dans un fichier .txt (comme j’ai eu du mal à filtrer depuis les 19000~ combinaison du coup j’ai pris les 304 combinaisons de l’internet) .

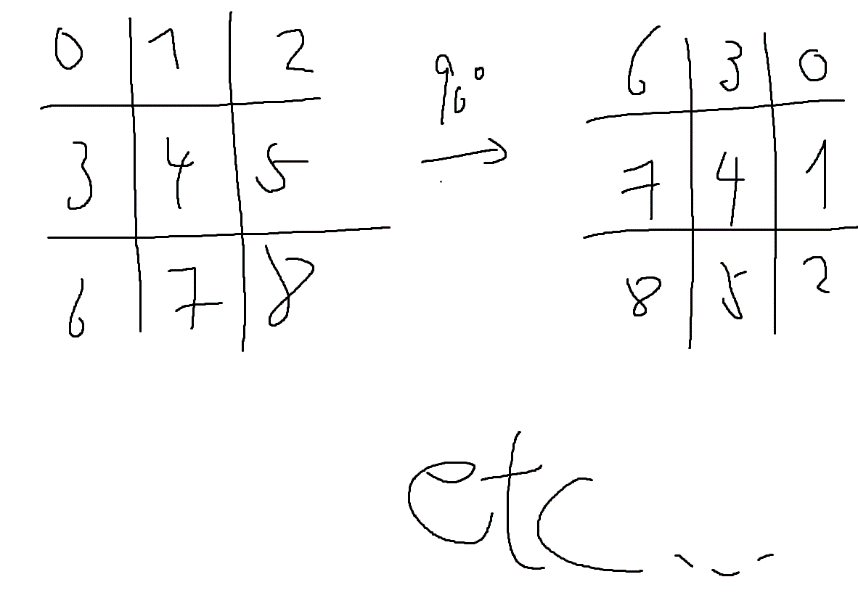
31/12/2021

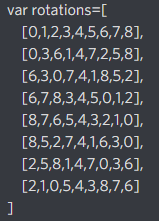
* Impossible de réaliser avec 304 car les configurations auront des billes de numéro différents :



Donc je vais faire un tableau de b10 avec tous les similaires donc aussi un tableau d’hachage de taille 304\*8 = 2432

* Changer d’idée, on garde les 304 combinaisons et on fait des rotations des 9 valeurs du tableau comme le grille avec la même fonction appliquer\_transformation()

Exemple :



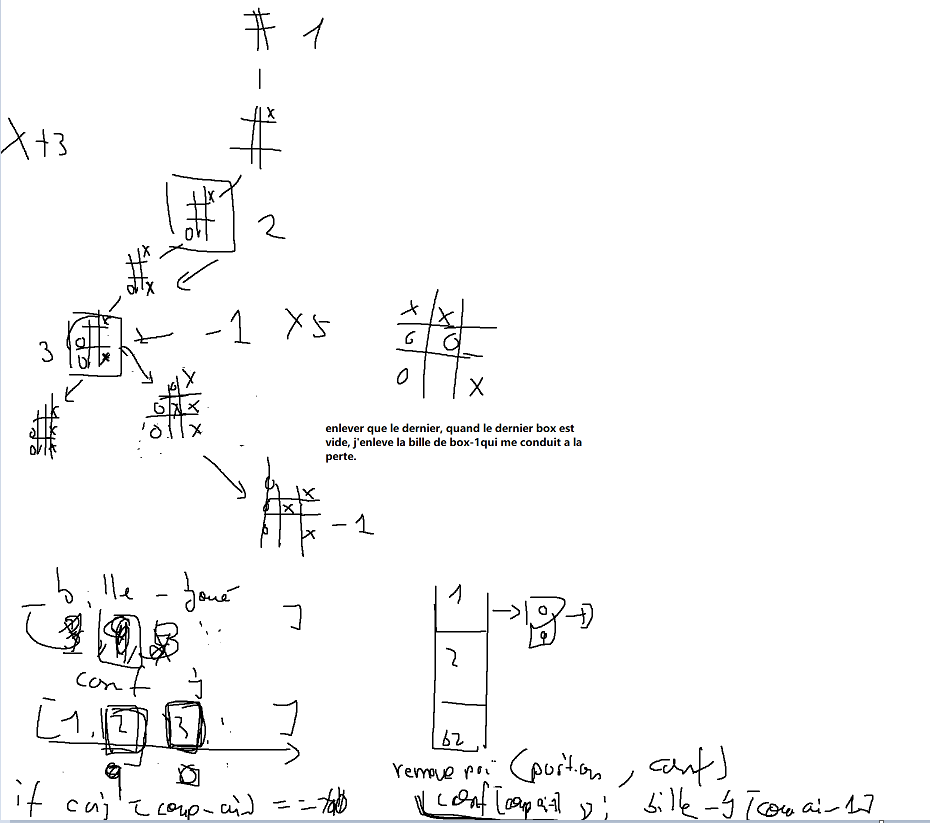
* Fini le mode joueur contre AI

01/01/2022

* Problème : certaines configurations ne sont pas inclues dans les 304 configurations j’ai fait un fichier avec 627 combinaisons
* Correction des bugs
* Finalisation de la fonction humain contre AI
* Commencement de la fonction AI vs AI
* Entrainement pas efficace et je n’arrive pas à trouver les problèmes

02/01/2022

* Après des printf j’ai trouvé qu’il s’agit un problème lors de la mise à jour des données. (Petit erreur de taper) Et l’entrainement fonctionne et devient efficace.
* Réalisation de l’interface (menu du jeu etc.) dans console.
* J’ai commenté tous les functions et codes
* Ce qu’il reste à faire :
  + Enregistrer le AI qu’on a entrainé. (la base de donnée, structure)
* Probleme de chemin (des box vide) du coup je vais faire les etape suivant(schema)



* Non reussi
* Limite du ai vs ai :  
  2 robot qui ne savent pas jouer au début donc il vont faire des etapes anormal pour gagner et ce la cree des problemes où le robot enregistrer le chemin dont qu’il pense ‘’gagnant’’

Le mieux est de laisser le robot jouer contre un robot qui est déjà pro pour apprendre les vrais chemin gagnant