

# Activité Hexapawn - Premier pas vers l'IA

Mr Tortillard

## Motivation

Beaucoup d'années sont passées entre la première discussion autour de la création d'une machine qui apprendrait à la manière d'un cerveau humain et l'accès à ChatGPT pour tous.

Ce TP va nous permettre de nous plonger légèrement dans le passé. Nous sommes au début des années 1960. Certains chercheurs réfléchissent à des programmes simple qu'un ordinateur pourrait apprendre afin de démocratiser le fonctionnement de l'apprentissage par une machine. Martin Gardner publie pendant 20 ans des articles dans le *Scientific American*. Ce magazine vise à vulgariser les concepts scientifiques au grand public. Parmi ces articles il écrit dans une section nommé *Mathematical Games* la description de différents jeux. Nous allons nous intéresser à un de ces jeu plus particulièrement, qui à permis de montrer comment une machine peut apprendre à jouer ...

## Hexapawn

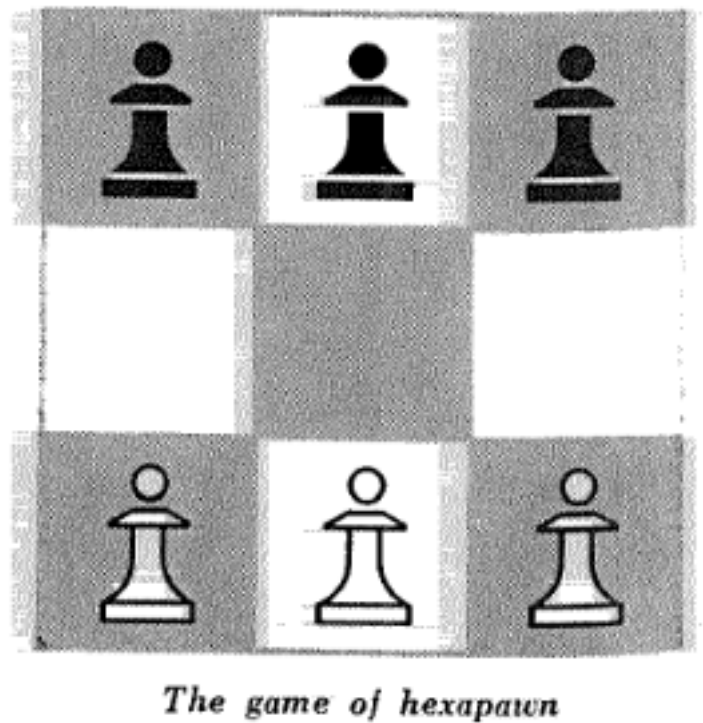


Figure 1: hexapawn

Hexapawn simule un jeu d'échecs en miniature.

On dispose d'un plateau de 3x3 cases sur lequel chaque joueur dispose de 3 pions.

Les règles de déplacement sont les mêmes qu'aux échecs.

A chaque tour deux possibilités s'offrent à nous pour déplacer un pion :

- Soit on avance. Un pion peut avancer si la case devant lui ne contient aucun autre pion.
- Soit on capture. Un pion capture d'une case devant lui en diagonale. Il ne peut capturer qu'un pion adverse.

On gagne une partie si une des 3 conditions suivante est rempli :

- Soit un de nos pions se trouve sur la ligne de départ de l'adversaire
- Soit les pions adverse ont tous étaient capturés
- Soit le dernier coup joué permet de bloquer l'adversaire. C'est à dire qu'il ne peut plus effectuer de coup.

## Intelligence artificielle

Voici la description de l'apprentissage d'une machine(sans électronique) du jeu Hexapawn par Martin Gardner.

La machine joue toujours les pions noirs. Cela implique qu'elle n'est jamais la première à jouer.

La machine à une liste de coup disponible pour chaque partie. Ces coups sont associés à une probabilité équivalente au début.

Tant que la partie n'est pas finie.

A chaque tour de jeu la machine regarde l'état de la partie puis choisit proportionnellement à leur probabilité, un mouvement valide.

Lorsque la partie est finie on a deux possibilités : - Soit la machine a perdu, dans ce cas rien ne se passe - Soit la machine a gagné, dans ce cas elle pondère à la hausse tout les mouvements qu'elle a choisis au cours de cette partie

La machine de Martin Gardner simulé ce procédé avec des boîtes d'alumette. Pour chaque boîte d'alumette

- sur le dessus des boîtes une configuration de partie est dessinée.
- dans chaque boîte il y a des billes de couleur différente chaque couleur représente un mouvement possible

# Activité

Notre but est de simuler cette machine en python.

Pour cela un dossier élève vous est fourni.

Ce dossier contient les classes qui simulent Hexapawn ainsi que la classe IA permettant de créer une machine qui apprend à jouer.

Dans un premier temps le but est de compléter les méthodes nécessaire à la simulation d'Hexapawn.

On commence par la classe Pion.

## Classe Pion

Un objet `Pion` possède un attribut `color` ainsi qu'une méthode `get_color(self)` permettant de récupérer la valeur de cet attribut.

De plus la méthode `__str__(self)` permet de représenter un pion selon sa couleur.

On utilisera les caractères associées au représentation des pions(à copier sur la page "*Symboles d'échecs en Unicode*" de wikipédia disponible ici).

### Compléter la classe Pion

## Classe Plateau

La classe `Plateau` est la classe qui s'occupe de toutes les règles relatives au jeu.

Avant de compléter cette classe, observez scrupuleusement chaque méthode ainsi que leur utilité (*lire la docstring et les doctests*).

**Voici les méthodes à compléter pour la classe Plateau:**

- `__str__(self)`
- `deplace(self, position_pion, action : str)`
- `est_gagnant(self, position_pion)`

## Classe IA

Jusqu'à maintenant les classes complétées servaient uniquement à représenter le jeu Hexapawn. La classe `IA` va permettre de simuler la machine de Martin Gardner avec des boites d'allumettes.

Une `IA` possède un attribut `self.boites` de type dictionnaire qui stocke les boites. Voici comment doivent être représentées les boites.

```
self.boites = {
    config1 : [ [(0,1,"G"),3], [(1,1,"A"),5], ...],
    config2 : [ [(2,0,"A"),1], [(0,0,"D"),1], ...],
    ...
}
```

où les `config` sont la représentation sous forme de chaîne de caractère d'un plateau(voir la méthode `__str__` de la classe `Plateau`).

La valeur associée à la clé `config` est une liste de `[mouvement_possible, poids]` où :

- Les `mouvement_possible` sont des triplets `(x_pos, y_pos, action)` tel que le pion se trouvant aux coordonnées `(x_pos, y_pos)` peut effectuer le mouvement `action`.
- La probabilité que `mouvement_possible` soit sélectionné est proportionnelle au `poids`.

**Voici le code à compléter pour la classe `IA`:**

- la **méthode** `ajoute_boites(self, pla : Pl, couleur_pion_qui_doit_apprendre = "n")`
- la **fonction** `choisis_au_hasard(liste_indices_mouvement_poids)`
- la **méthode** `pondere_hausse(self, couple_config_mouvement_choisis_par_IA : list[tuple])`