



IAP4

**Intelligence artificielle de puissance 4**

- DaemonLab -  
pedagogie@ecole-89.com

*Ce document est strictement personnel et ne doit en aucun cas être diffusé.*

## Table des matières

Détails administratifs.....	3
Énoncé.....	4
API logicielle de jeu.....	4
Entraînement des individus.....	4
Lancement de parties.....	5
Bonus.....	5

## Détails administratifs

Votre travail doit être **envoyé sur la [forge logicielle](#)**. Le nom de votre dépôt doit être **2021\_iap4**. Un nom de dépôt erroné donnera lieu à un échec du TP.

Vous devez donner le droit en lecture à l'utilisateur **delivery-collector**, qui sera chargé de ramasser votre travail pour le corriger.

Vos identifiants de connexion à la [forge](#) sont votre mot de passe LDAP/UNIX, tel qu'il fonctionne sur les postes de l'école.

Médailles accessibles :



**Rendu Git :**

Vous avez réussi à rendre votre travail sur la forge logicielle

## Énoncé

Vous devez implémenter dans ce projet :

- une API logicielle de jeu pour le « puissance 4 » ;
- un entraîneur de réseau de neurones, pour ce jeu, qui fasse usage de votre interface ;
- un système pour lancer des parties entre IA ou entre humain et IA.

### API logicielle de jeu

Votre interface de jeu doit comporter :

- une structure de jeu, contenant les informations relatives à une unique partie (soit l'état du plateau, à quel jour le tour, si la partie est finie, à quel joueur la victoire, etc.) ;
- une fonction de création de jeu ;
- une ou deux fonction-s pour faire jouer un joueur ou l'autre ;
- une fonction pour afficher le plateau de jeu sur le terminal ;
- (en bonus) une fonction pour afficher le plateau dans un tampon de pixel pour la libla-pin ;
- une fonction pour détruire le jeu et les ressources allouées lors de la création du jeu.

Vous devez prévoir un mécanisme d'erreur dans le cas où un joueur tente de jouer à un endroit impossible.

À la fin d'une partie, il ne doit pas être possible de continuer à jouer.

### Entraînement des individus

Vous devez programmer un système capable d'entraîner des générations d'individus au jeu. Il doit se servir de l'API (de l'ensemble de vos fonctions + la structure de jeu) décrite au-dessus pour instancier des parties entre des individus, et tester leur capacité à jouer correctement et à gagner.

Un bon individu respecte les règles, un individu performant peut gagner des parties.

Les meilleurs individus de votre série des tests doivent être sérialisés dans des fichiers (un fichier par individu). Votre fonction d'entraînement doit pouvoir désérialiser des individus à partir d'un fichier afin de commencer les tests avec eux. Sans individus sources, l'entraînement est commencé avec des individus aléatoires.

## Lancement de parties

Vous devez faire une fonction qui lance et pilote une partie soit avec deux IA, soit avec une seule. Dans ce deuxième cas, le deuxième joueur sera humain et aura une interface de jeu sur la sortie standard. (Que vous devez implémenter.)

Pour les intelligences artificielles, vous devez passer en paramètre à ces fonctions des chaînes de caractères représentant des chemins vers les fichiers dans lesquels vous les avez sérialisées.

## Bonus

Votre entraîneur d'IA se sert des threads POSIX pour effectuer plusieurs travaux en même temps.

Vous avez fait une interface de visualisation avec la liblapin.

Vous avez un jeu en réseau, qui fonctionne entre deux ordinateurs différents.