Séance 1

1) Préparation de l'environnement de développement

a) Système d'exploitation

Vous devrez travailler sur Linux ou MacOS. Si votre ordinateur dispose actuellement seulement de Windows, des solutions sont :

- installer une VM Ubuntu sur Windows : recommandé

- installer Linux en dual boot : recommandé seulement si vous êtes à l'aise avec l'installation d'OS

- installer Windows Subsystem for Linux : plus complexe et moins ergonomique que VM

- travailler à distance sur une machine Linux de l'ENSTA : le plus simple, mais la latence réseau peut etre embetante et l'installation ENSTA de conda est actuellement (le 27/04/21) buggée

b) Environnement python

On suppose ici que vous êtes sur une machine Linux ou MacOS.

1. Installation de conda
- Pour les personnes travaillant sur une machine de l'ENSTA, executer la commande suivante pour disposer de conda :

```
echo 'useensta python3' >> ~/.bashrc ; source ~/.bashrc
```

- Pour les autres, installer anaconda sur votre machine si ce n'est pas déjà fait.

2. Ensuite :
- Créer un environnement conda nommé «in104» avec python 3.7 en éxecutant la commande:
  - `conda create --name in104 python=3.7`
- A chaque fois que vous ouvrirez un nouveau terminal, pensez à éxecuter :
  - si la version de conda est <4.4 (comme sur les machines ENSTA) : source activate in104
  - si la version de conda est ≥ 4.4 : conda activate in104

  Vous trouverez votre version de conda avec la comande : conda --version
- Si vous souhaitez ne pas avoir besoin d'éxecuter cette commande à chaque ouverture de terminal, vous pouvez la rajouter à la fin de votre fichier ~/.bashrc.

2) Création du dépôt git

Comme pour les TDs précédents, vous utiliserez un dépôt git sur Github.com pour partager votre travail avec votre binôme et moi-même.

Plutôt que chaque groupe ait sa propre arborescence de code python, j'ai créé un template de projet, duquel vous partirez comme point de départ. Cela permettra que tous les groupes aient la même structure de projet et me permettra de facilement récuperer votre code pour la compétition inter-groupes.

1) Pour cela, l'un des membre du binome doit :

- se rendre sur https://github.com/clement-masson/IN104_PROJECT_TEMPLATE
- cliquer sur le bouton vert « Use this template »
- choisir un nom de repo de la forme IN104_PROJECT_NOM1_NOM2
- cocher l'option «private» puis cliquer sur « create repository »
- se rendre dans les paramètres pour ajouter votre binôme et moi-même (clement-masson)

2) Une fois le dépôt en ligne créé, l'un d'entre vous doit :

- le cloner sur sa machine
- renommer les dossiers IN104_PROJECT_TEMPLATE en IN104_PROJECT_NOM1_NOM2.
- modifier de même « IN104_PROJECT_INSERT_YOUR_GROUP_NAMES_HERE » dans le fichier setup.py à la racine de votre dépôt.
- commit et push ces changements. Le second membre du groupe peut alors cloner à son tour ce dépôt sur sa machine.

3) Une fois le dépôt cloné sur sa machine, chaque membre doit « installer » ce package python à l'aide de la commande (en étant placé dans le dossier contenant le fichier setup.py) :

```
pip install -e ./
```

Pour la culture générale: ici, « installer » signifie simplement copier les fichiers du package aux bons endroits sur votre pc pour que python trouve vos modules lorsque vous executerez « import IN104_PROJECT_NOM1_NOM2.mon_module »dans un autre module python toto.py situé n'importe ou sur votre pc (ie pas besoin que toto.py soit placé dans le même dossier que mon_module.py pour qu'il puisse utiliser mon_module.py).

3) Le package aiarena

Le package aiarena est un package sachant gèrer le déroulement d'une partie, vérifier que les coups choisis par les joueurs sont valides et afficher l'état du plateau de jeu après chaque coup. Le package fournit également les fonctions permettant de calculer les coups possibles et états accessibles depuis un état de jeu donnée. Cela vous permet de vous concentrer sur l'implémentation de votre IA (i.e le choix du coup) plutot que de celle de la recherche des coups. Les jeux actuellement disponibles dans le package aiarena sont les dames, les echecs, puissance4 et abalone.

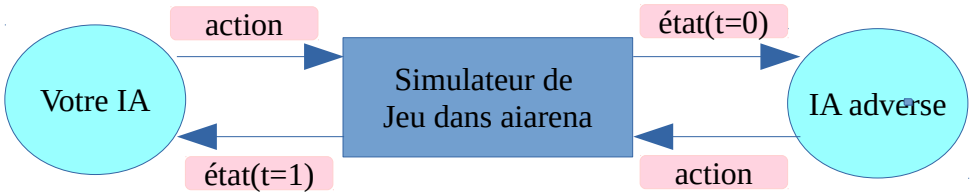

Illustration 1: Integration entre simulateur et IAs.

Le simulateur envoie un état de jeu à une IA, qui renvoie le coup choisi. Le simulateur vérifie que le coup est valide, calcul le nouvel état et l'envoie à l'autre IA, qui à son tour choisit un coup.

Le projet consistera à implémenter une classe disposant d'une méthode *play* qui prendra en entrée un état de jeu accompagné d'un temps limite d'exécution et devra renvoyer dans cette durée limitée le coup choisi.

a) Installation

3) Ce package est public et disponible sur Github. Installez-le avec la commande suivante :

```
pip install git+https://github.com/clement-masson/aiarena.git@master
```

Si tout s'est bien passé, le terminal devrait conclure par :

```
Successfully installed aiarena-1.0
```

b) Description rapide

Une description plus détaillée du package aiarena est acessible sur le wiki du package.

Brievement, les classes que vous aurez à utiliser dans ce package sont :

- Game : elle permet de lancer une partie. Son constructeur prend comme argument : le type de jeu, suivi des Ias et de leur temps imparti pour chaque coup.

Pour chaque type de jeu :

- GameState : contient l'état d'une partie à un instant donné.

- Move : représente un coup possible dans un état de jeu donné.

- Cell : représente le contenu d'une case du plateau de jeu.

c) Test et démonstration

Tel qu'il est fournit, le package aiarena contient le nécessaire pour jouer entre deux humains (attention cependant: l'interface est non graphique et rudimentaire!).

4) Completez dans votre package le fichier **scripts/human_vs_human.py** afin de lancer une partie entre deux humains sur un des quatre jeux disponibles. Pour cela, suivez les instructions présentes sur le wiki.

Pour executer ce script, entrer la commande depuis n'importe quel repertoire:

```
python -m IN104_PROJECT_NOM1_NOM2.scripts.human_vs_human
```

4) Création de votre première «IA»

Le but est ici de vous faire coder une IA très rudimentaire afin de vous familiariser avec le package aiarena.

> Pour la question suivante, vous pourrez vous inspirer du code source de ManualBrain du package AIArena. Faites cependant attention à ne garder que le stricte nécessaire.

5) Dans le fichier randomBrain de votre package, implémentez la classe *RandomBrain* dont la méthode *play* sélectionne au hasard un des coups possibles. Vous pourrez pour cela vous aider du module *random* de Python ainsi que des méthodes fournies par *GameState*. Pendant qu'un d'entre vous fait cela, l'autre membre du groupe pourra faire la question suivante.

6) Completer le fichier **scripts/random_vs_random.py** de votre package afin de lancer une partie entre deux RandomBrain sur chacun des 4 jeux disponibles et affiche les PGN après chaque partie.

Cette page wiki pourra vous être utile.

Pour executer ce script, entrer la commande depuis n’importe quel repertoire:

```
python -m IN104_PROJECT_NOM1_NOM2.scripts.random_vs_random
```

> De manière générale, le wiki contient un certain nombre d’informations sur le package AIArena. Si vous avez une question sur son fonctionnement pour laquelle vous n’y trouvez pas la réponse, demandez-moi.