



TP n °5: Puissance 4

Durée : 4H

Jouer en réseau et finalisation du jeu

Le but de ce TP est de permettre aux programmes de jouer en réseau. Pour cela un serveur a été installé et des fonctions écrites en python ont été livrées. Elles permettent de :

- Créer une partie et récupérer le numéro de la partie
creerPartie() : entier
- Rejoindre une partie à partir de son numéro
rejoindrePartie(entier : numéro de partie) → booléen : bien rejoint
- Transmettre un entier correspondant à un numéro de colonne et recevoir le coup de l'adversaire
jouer(entier : mon coup) → entier : coup de l'adversaire
- Attendre le coup de l'adversaire (à ne faire qu'une seule fois lorsque l'on rejoint une partie)
attentePremierCoup() → entier : coup de l'adversaire
- Terminer une partie en annonçant le dernier coup (sans attendre de réponse de l'adversaire) et récupérer le temps dépensé en seconde
fin(entier : coup final) → entier : temps dépensé

Ces fonctions sont données dans le fichier intitulé competition.py (si vous n'avez pas la fonction fin, veuillez télécharger le nouveau fichier competititon.py sur arche).

Voici le protocole de communication associé

Client 1	Client 2
id=creerPartie()	
	rejoindrePartie(id)
	Colonnejoueparclient1=attentePremierCoup()
Colonnejoueparclient2=jouer(X)	
	Colonnejoueparclient1=jouer(X)
Colonnejoueparclient2=jouer(X)	
...	Colonnejoueparclient1 (X)
	temps = fin(X)
temps = fin(0)	

Il est à noter que le serveur ne gère rien d'autre que les coups joués. Aucun damier n'est reconstitué au niveau du serveur donc aucune détection de validité ou fin de partie est gérée. *L'indice de la première colonne est 0.*

Concernant le **Gameplay**, voici les fonctionnalités que devra proposer votre programme :



Au lancement de la partie le programme soit laisser 3 possibilités :

Voulez-vous jouer :

- A 1 joueur contre l'IA ? ;
- A 2 joueurs humains ?;
- Avec votre IA en réseau contre une autre IA ?

Cas 1 : 1 joueur contre IA :

Le programme demande le pseudo du joueur, et demande qui commence. La partie commence ensuite. Le premier joueur commence toujours par une croix.

Cas 2 : 2 joueurs humains

Le programme demande le pseudo des 2 joueurs, et demande lequel commence. Le premier joueur commence toujours par une croix.

Cas 3 : Une IA qui joue en réseau

Le programme demande s'il faut créer la partie ou rejoindre une partie existante.

S'il faut créer la partie, alors le programme affiche dans la console un numéro de partie qui sera communiqué oralement à son adversaire. Le programme qui créé la partie commence systématiquement, et avec une croix.

S'il faut rejoindre une partie, alors le programme demande via la console un numéro de partie (celui-ci est communiqué oralement par son adversaire). Le programme qui rejoint la partie attend systématiquement le premier coup joué par l'adversaire. L'adversaire ayant une croix, celui qui rejoint joue avec les ronds.

Il ne doit avoir aucune autre question ou interruption dans le mode « IA qui joue en réseau ». Le damier doit simplement être affiché automatiquement à chaque coup joué et à chaque coup reçu.

Si l'adversaire joue un coup non valide, votre programme doit afficher « l'adversaire ou joué la colonne X qui n'est pas un coup valide. Il a donc perdu la partie. Partie terminée.»

La fin de partie en réseau :

Si vous détectez la fin d'une partie lorsque vous jouez votre prochain coup, alors plutôt que d'envoyer votre coup à l'aide la fonction jouer (qui attend une réponse), vous devez envoyer le dernier coup à l'aide la fonction fin qui vous donnera le temps dépensé par votre IA.

Si à l'inverse, vous détectez qu'il y a une fin de partie à la réception du coup de votre adversaire, utiliser la fonction : fin(0) qui vous donnera le temps dépensé par votre IA.

Attention : utiliser la fonction fin alors que ce n'est pas vraiment la fin est éliminatoire au même titre que de jouer un coup non valide.

Aide pour ceux qui sont perdus :

Si la variable mon_coup contient le prochain coup que vous voulez jouer et D votre damier, la brique de base minimale de votre programme doit contenir :

```
empiler(D, mon_coup, 'X')  
coup_adverse = jouer(mon_coup)  
empiler(D, coup_adverse, 'O')
```



Barème de Notation : respect des consignes sur 12 points :

La séance est libre. A l'issue de celle-ci les programmes devront pouvoir :

-Afficher le damier	1 point
-Permettre à 2 joueurs humains de jouer alternativement	1 point
-Détecer les cases non valides	1 point
-Détecer les conditions de victoires	2 points
-Détecer le match nul	1 point
-Proposer de rejouer une partie	1 point
-Jouer contre l'intelligence artificielle en mode un joueur	2 points
-Avoir un gameplay conforme avec ce qui est affiché ci-dessous	3 points
-Le fichier le plus volumineux ne dépasse pas 100 lignes et le code est lisible	1 point

Les étudiants qui créeront une interface graphique (ajout de sons, d'images etc.) pourront se voir offrir jusque 4 points de bonus.

Les 8 derniers points seront attribués sur le résultat de l'IA lors de la compétition du 12 février. 8 points pour le vainqueur, 0 point pour le dernier. Les modalités de la compétition seront données en séance.