

# Projet AI Quarto

---

Pour notre travaille de fin de 2ème Bac en ingénieur industriel a l'Ecam orientation Génie Electrique, notre professeur d'informatique nous a mis au défis de créer une intelligence artificiel jouant au jeux Quarto.

Pour ce faire il nous a mis a disposition une librairie ainsi qu'un code pour la structure du jeux.

## Fonctionnement

---

Pour créer mon intelligence artificiel j'ai décidée d'utiliser la librairie EasyAI <http://zulko.github.io/easyAI/index.html> grace a cette librairie j'ai pu utiliser différent algorithme du type Alpha-Beta pruning, dans mon cas j'utilise plus précisément l'algorithme Negamax <https://en.wikipedia.org/wiki/Negamax> ainsi que la méthode Solving qui va résoudre la partie en utilisant Negamax avec différente profondeur de recherche.

Pour générer le coups qui sera jouer différente intelligence sont disponible ( voir **Intelligence** ).

L'intelligence principale, l'intelligence *client*, utilise la fonction `id_solve` de la class Solving qui va me retourner plusieurs information dont la mouvement le plus interessant pour arriver a la victoire le plus rapidement.

## Utilisation

Pour jouer au jeux vous devez lancer 3 terminale depuis le dossier ou est enregistré le jeux.

1. [Fenêtre 1] : server du jeux
  2. [Fenêtre 2] : client 1 du jeux
  3. [Fenêtre 3] : client 2 du jeux
- vous pouvez lancer 2 fois le même client*

## Intelligence

1. client : utilisation de `id_solve` (*recommander*)
2. clientB : utilisation de Negamax avec transposition table

3. user : pour jouer contre l'AI
4. rdm : AI qui agit 100% aléatoirement
5. prof : AI d'origine

## Lancer une partie

### Server :

```
./quarto.py server --verbose
```

### Client :

```
./quarto.py *Intelligence* *Nom* --verbose
```

Vous avez aussi la possibilité de lancer les clients et server sur différente machine, il vous faudra donc préciser l'ip du host ainsi que le port de communication (*Pars défaut le host = localhost et le port = 5000*) :

### Server distant :

```
./quarto.py server --verbose --host= *IP* --port= *Port*
```

### Client distant :

```
./quarto.py *Intelligence* *Nom* --verbose --host= *IP* --port= *Port*
```