

---

### Projet Saison 3

#### Vue d'ensemble

Pour votre projet final, vous allez concevoir et mettre en œuvre des algorithmes d'intelligence artificielle (IA) permettant de jouer à des jeux simples. Ces algorithmes seront testés et notés par rapport à des algorithmes factices ainsi qu'aux soumissions d'autres étudiants. Votre intelligence artificielle sera simple au début, mais à chaque itération, elle doit s'améliorer et s'adapter aux nouvelles exigences et aux adversaires de plus en plus difficiles.

La troisième saison vise à concevoir une intelligence artificielle compétente capable de vaincre les adversaires de force moyenne. Les élèves apprendront un algorithme de comptage de fréquence en tant que bonne stratégie de départ. Les étudiants qui souhaitent se défier davantage peuvent tenter de battre les autres élèves dans le classement pour obtenir des points bonus.

#### Distribution des notes

Jeu 1 (RPSLS)	10%
Jeu 2 (à annoncer plus tard)	10%
Rapport	10%
<b>Total</b>	<b>30%</b>

Saison 3 - Date limite de la soumission est le 12 mars (bonus)

#### Instructions

1. Cloner le projet à <https://github.com/felixsoum/420J13AS-RPSLS>
2. Recherchez le fichier qui vous est attribué. Il sera situé dans le chemin:  
    \RPSLS\AI\S{numéro de section}\{4 lettres du code permanent}.cs
3. Fix any problems from the previous week.
4. Substituez (override) et implémentez la méthode `Observe()` qui permet à votre IA d'observer le coup précédent de votre adversaire.
5. Mettez à jour votre méthode `Play()` pour prendre en compte ces observations afin de surmonter les défis liés à l'IA.
6. Configurez votre IA en allant sur `Program.cs`, en remplaçant `RandomAI` par 15, à la ligne 15, puis en exécutant le programme (Ctrl + F5).

```
var game = Game.Create<RandomAI>();
```

7. Soumettez seulement votre dossier de classe à Lea (XXXX.cs).

## Règles

La règle la plus importante est de ne pas plagier. Il est recommandé et bon de travailler avec des camarades de classe pour discuter de stratégies et de pratiques opposées, mais copier-coller le travail de quelqu'un d'autre est du plagiat. En tant que informaticiens, nous avons conçu de très bons outils pour détecter cela dans le code, même si les variables sont renommées ou les instructions réorganisées. Vous pourriez même apprendre quelques algorithmes pour cela dans ce cours. En gros, ne faites pas cela ou vous serez fortement pénalisé. De plus, vous ne pouvez pas dépendre d'autres classes d'intelligence artificielle (n'essayez pas d'invoquer les méthodes d'un autre). Vous serez également pénalisé si vous faites cela.

Les règles suivantes sont des directives pour le tournoi. Si vous ne les suivez pas, vous serez disqualifié, mais vous ne serez pas pénalisé pour les notes du projet.

- De nouvelles règles ne seront ajoutées qu'après la fin d'un tournoi et suite à la détection de comportements indésirables.
- Jouer un coup illégal entraînera la perte du tour.
- Si vous avez besoin d'un nouvel objet aléatoire, vous êtes uniquement autorisé à utiliser `Game.SeededRandom`, qui est initialisé car les résultats du jeu doivent être reproductibles (déterministes).
- Chaque tour de votre IA doit prendre moins de 10 millisecondes pour être exécuté. Un tour consiste en un appel à `Observe()` puis à `Play()`. Si votre IA prend trop de temps, le système vous le dira dans la console. Ce délai est sujet à changement.
- Vous n'êtes pas autorisé à utiliser les classes de l'espace de noms `System.Reflection`, car vous n'en avez pas besoin et si vous le faites, je me méfie de votre code.
- Si vous essayez d'instancier vous-même une intelligence artificielle, vous obtiendrez une exception non autorisée.

## Défis

Chaque défi met votre IA contre une IA factice spécifique. Un défi est composé de 20 batailles et chaque bataille est composée de 100 rondes de RPSLS. Pour réussir le défi, votre IA doit vaincre le mannequin dans toutes les batailles, mais pas nécessairement à toutes les rondes. Une bataille est gagnée par l'IA qui a remporté le plus grand nombre de rounds. Entre chaque combat, la graine aléatoire (random seed) changera. La graine initiale (initial seed) sera également modifiée avant d'évaluer vos IA. Cela rend impossible à une IA de réussir un défi en raison de la chance ou du codage difficile de tous les mouvements. Vous pouvez basculer les défis simplement en décommentant ou en commentant le code (`Program.cs`, ligne 17-19). Vous pouvez également exécuter plusieurs défis de manière séquentielle.

```
game.Challenge<FavoriteOneAI>();  
game.Challenge<FavoriteTwoAI>();  
game.Challenge<RepeaterAI>();
```

## Classement

Un tournoi est mis en place pour encourager les étudiants à se surpasser. Les meilleurs élèves de chaque section auront la possibilité de présenter de manière informelle leur intelligence artificielle à la classe afin de recevoir des points bonus. N'hésitez pas à essayer le tournoi vous-même en exécutant l'instruction `game.PlayTournament()` dans la ligne 21 de `Program.cs`. Notez que si vous souhaitez obtenir le même résultat que le tournoi réel, veuillez à désactiver (commenter) les défis, car ils affecter le résultat par l'utilisation de l'objet `SeededRandom`. La graine (seed) sera tirée des numéros de loto hebdomadaires

```
game.PlayTournament();
```

## Le pointage

**[1 points]** Pour avoir réussi le défi contre *FavoriteOneAI*. Cette IA sélectionnera son coup favori lorsqu'elle sera instanciée, puis elle est assuré de jouer ce coup au moins 40% du temps (minimum). Chaque autre coup est garanti d'être joué au plus 15% des fois (maximum).

### **[1 points]**

Pour avoir réussi le défi contre *FavoriteTwoAI*. Cette IA choisira une paire de coups favoris lorsqu'elle sera instanciée, dans cette liste:

- Rock & Spock
- Paper & Lizard
- Scissors & Rock
- Spock & Paper
- Lizard & Scissors

Ensuite, elle est assuré de jouer un coup de la paire choisie au moins 40% du temps (minimum). Chaque autre coup est garanti d'être joué au plus 15% des fois (maximum).

### **[1 points]**

Pour avoir réussi le défi contre *RepeaterAI*. Cette IA choisira une séquence aléatoire de coups d'une longueur comprise entre 20 et 39 lorsqu'elle sera instanciée. Ensuite, elle exécutera les mouvements qui suivent cette séquence, puis reviendra au début en boucle et recommencera.