

Projet d'Intelligence Artificielle

MP :XMS1IE042

Alexandre Bruckert

alexandre.bruckert@univ-nantes.fr

8 novembre 2024

Nantes Université

Description du jeu

Pour ce projet, on considère un jeu simple à deux joueurs : le jeu Isolation. Chaque joueur est placé dans un coin d'une grille carrée de taille $n \times n$.

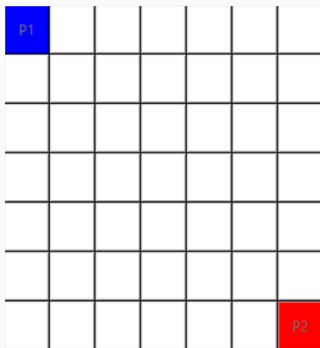


Figure 1 : Un exemple de grille de départ, avec $n = 7$

Description du jeu

L'objectif pour chaque joueur est de faire en sorte que le joueur adverse n'ait plus de mouvement possible.

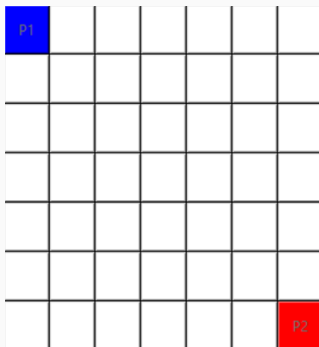


Figure 1 : Un exemple de grille de départ, avec $n = 7$

Description du jeu

Le joueur 1 commence. A chaque tour, un joueur doit effectuer deux actions successivement :

- Déplacer son pion sur une case adjacente libre (orthogonalement ou diagonalement)
- Bloquer une case libre, qui deviendra inaccessible jusqu'à la fin de la partie.

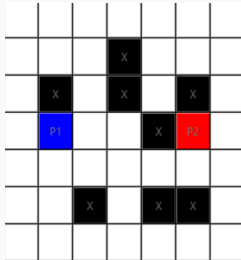


Figure 2 : Exemple d'une partie en cours

Description du jeu

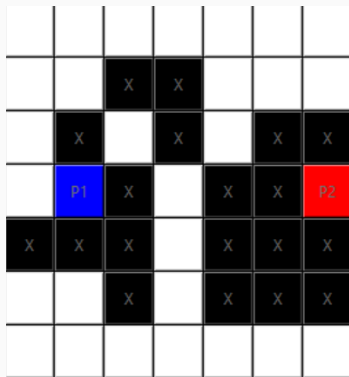


Figure 3 : Exemple d'une partie terminée

Objectifs du projet

Dans ce projet, vous devrez proposer et implémenter une (ou plusieurs!) stratégie pour ce jeu. Pour ce faire, vous trouverez sur Madoc et Gitlab le code permettant de modéliser le jeu (classes implémentant le jeu, UI, etc). Vous

Dans la suite, nous considérerons toujours le jeu avec $n = 9$. Cependant, n'hésitez pas à réduire cette taille lors de vos tests!

Note importante : Sauf autorisation préalable, vous ne devrez utiliser aucune librairie autre que celles déjà importées dans le code fourni!

Description du projet

Ce projet sera divisé en 3 parties.

- Partie 1 : Commencer par le commencement.
 1. Familiarisez vous avec le code fourni dans les différents fichiers. Assurez vous de bien comprendre le fonctionnement de chaque classe, ainsi que le fonctionnement général.
 2. Implémentez une stratégie jouant contre un être humain de manière aléatoire.
 3. Concevez et implémentez une procédure de comparaison permettant d'évaluer la supériorité d'une approche sur une autre. Pour ce faire, vous devriez faire jouer plusieurs fois les stratégies, en inversant le joueur commençant à chaque fois. Vous implémenterez cette procédure dans un nouveau fichier `evaluate.py`.

Description du projet

- Partie 2 : Ajouter de l'intelligence.
 1. Implémentez une stratégie MinMax permettant d'améliorer les performances du joueur aléatoire. Vous ferez particulièrement attention à la profondeur autorisée dans la recherche!
 2. Modifiez cette stratégie en utilisant un élagage $\alpha - \beta$

- Partie 3 : Améliorer les performances.
 1. Proposez et ajoutez des heuristiques permettant d'améliorer les performances. Vous ferez attention de bien justifier et décrire vos choix.
 2. Implémentez une stratégie de recherche de Monte Carlo. Décrivez cette approche, ainsi que les différences avec les approches précédentes en termes de performances.
 3. (Bonus) Si il vous reste du temps, vous pouvez considérer des modèles d'apprentissage pour la création d'une fonction d'évaluation (Q-learning, par exemple).

Consignes

- Pour ce projet, vous travaillerez en **binômes**.
- Vous commencerez par **cloner** (ou mieux : **forker**) le dépôt sur lequel vous trouverez le code :

```
git clone https://gitlab.univ-nantes.fr/bruckert-a/isolation_game
```

- A la fin de chaque séance, vous devrez **commit et push** les avancées que vous aurez faites.
- Vous n'avez (normalement) que le fichier **strategy.py** à modifier. Si vous voulez implémenter des fonctions utiles supplémentaires, faites-le de préférence également dans ce fichier.

- Vous soumettrez sur Madoc un document pdf créé avec \LaTeX , décrivant et expliquant en détails votre travail.
- Ce document devra contenir un lien vers un dépôt Gitlab public, où vous aurez déposé l'intégralité de votre code.
- Vous devrez également, dans ce document, répondre aux questions préliminaires à la fin de ces slides.
- Ce projet est à rendre pour le vendredi 10 janvier 2024, mais une stratégie fonctionnelle sera attendue pour la dernière séance, semaine 51.

Le projet sera évalué selon plusieurs critères :

- La qualité et la complétude du rapport ($\sim 35\%$)
- La qualité de la solution proposée et de son implémentation ($\sim 35\%$)
- Les performances de la solution proposée lors de la dernière séance et sur un benchmark ($\sim 15\%$)
- La présentation de votre solution lors de la dernière séance. ($\sim 15\%$)

Il est obligatoire d'être présent à au moins 4 séances sur 6, en plus de la dernière séance. 5 points par séance supplémentaire manquée seront retirés de la note finale.

- Vous pouvez vous aider de LLMs pour développer votre code à la condition que vous puissiez comprendre et justifier chacun des choix techniques.
- Vous pouvez également vous aider de LLMs à des fins de relecture et correction pour votre rapport.
- Dans tous les cas, vous devez indiquer votre utilisation de LLM dans chaque document en ayant bénéficié.
- Vous êtes responsables de ce que vous rendez!

Questions préliminaires

1. Une partie peut-elle se terminer par un match nul ? Pourquoi ?
2. Caractérisez le jeu en termes de (i) son information disponible ; (ii) son caractère aléatoire
3. Exprimez une borne supérieure sur la complexité de l'espace des états du jeu (c-à-d le nombre de positions possibles pour le jeu) en fonction de n
4. Le **facteur de branchement moyen** d'un jeu est le nombre moyen de branches filles pour chaque noeud parent de l'arbre d'état du jeu. Donnez une estimation (même grossière) du facteur de branchement moyen pour ce jeu.
5. Que peut-on en conclure ?