Intelligence Artificielle (IA)

Les jeux, recherche totale

Akka Zemmari

LaBRI, Université de Bordeaux

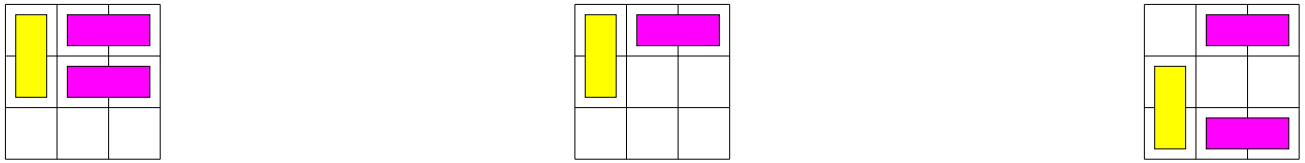
2024 - 2025

Un problème simple

Les dominos de couleurs

Principe

Sur une grille $n \times n$, on doit être le dernier à poser son domino 2×1. Violet joue horizontalement et jaune verticalement.

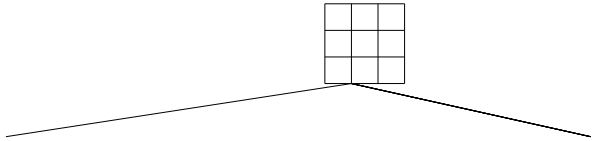


Un jeu simple mais illustratif

Nous allons utiliser ce jeu simple pour schématiser les approches

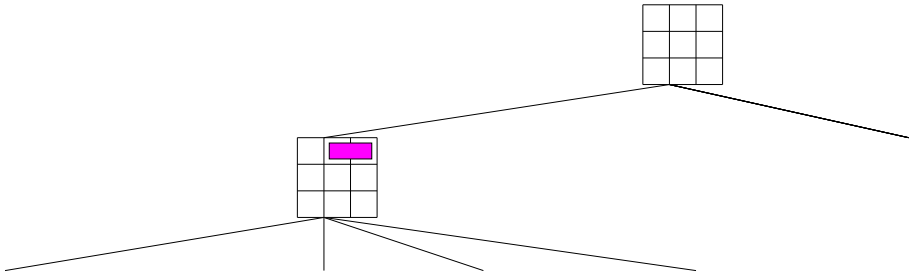
Exploration de l'arbre de jeu

La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Exploration de l'arbre de jeu

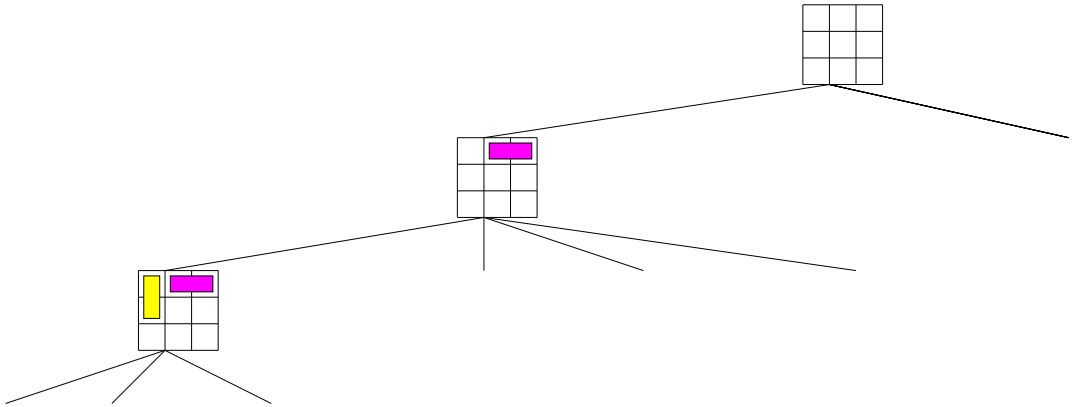
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

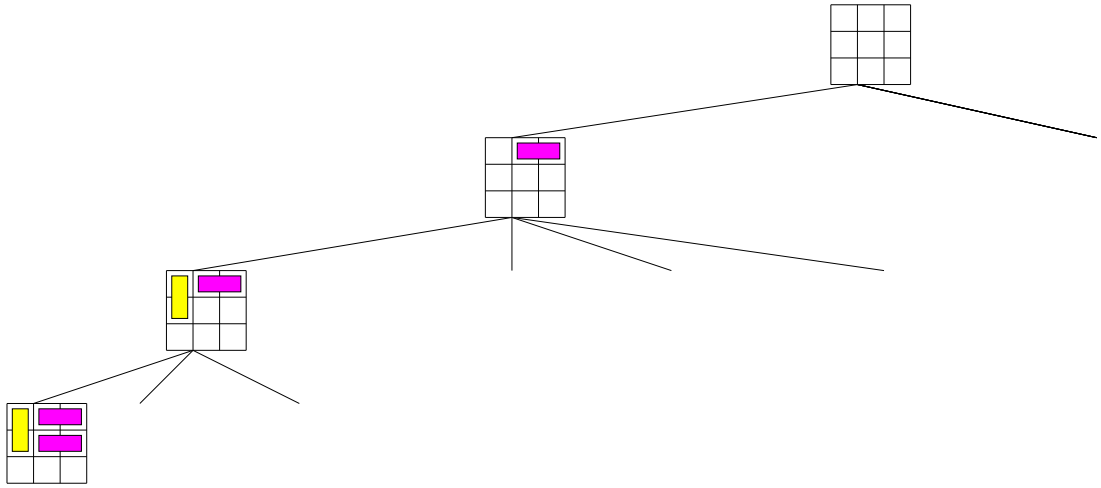
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

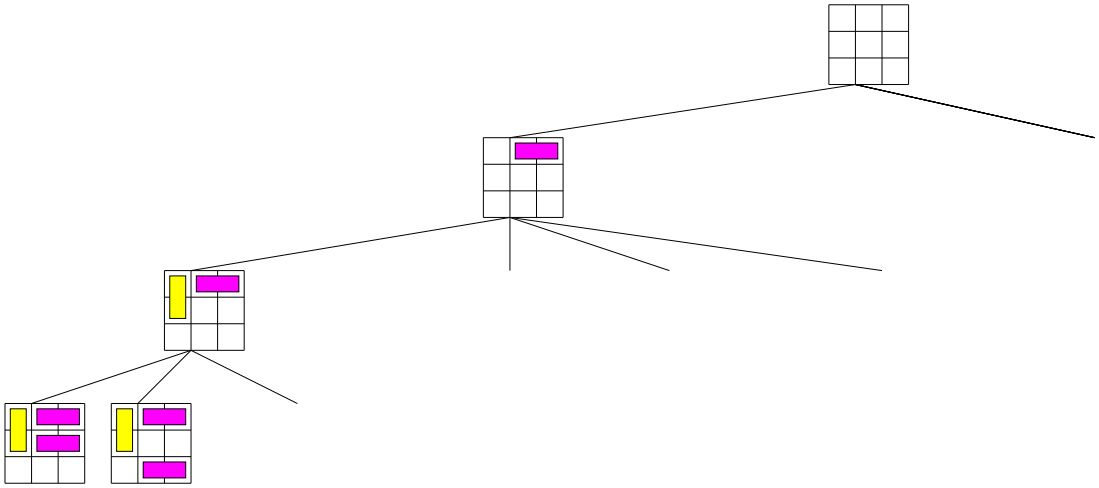
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

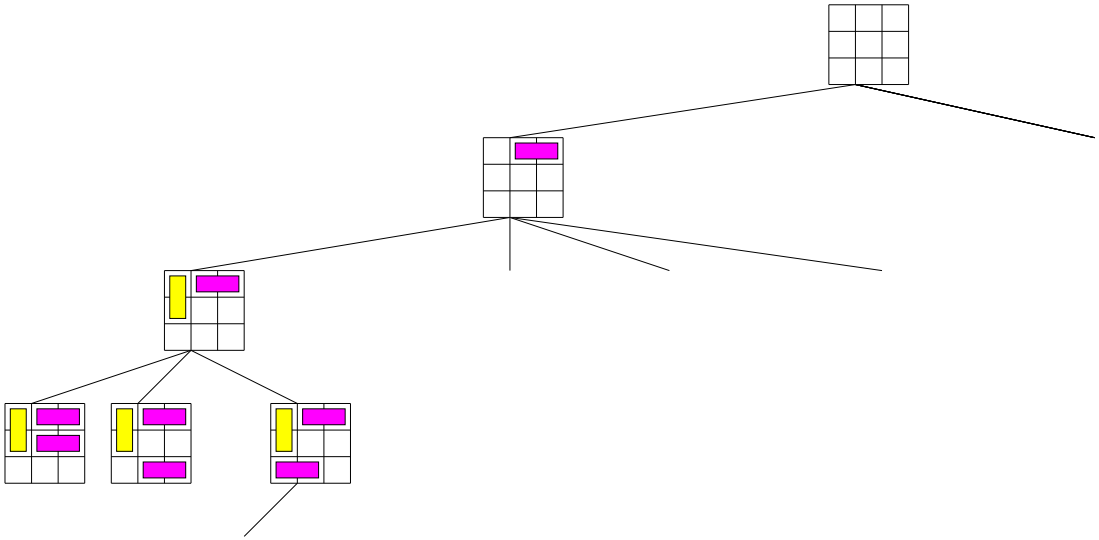
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

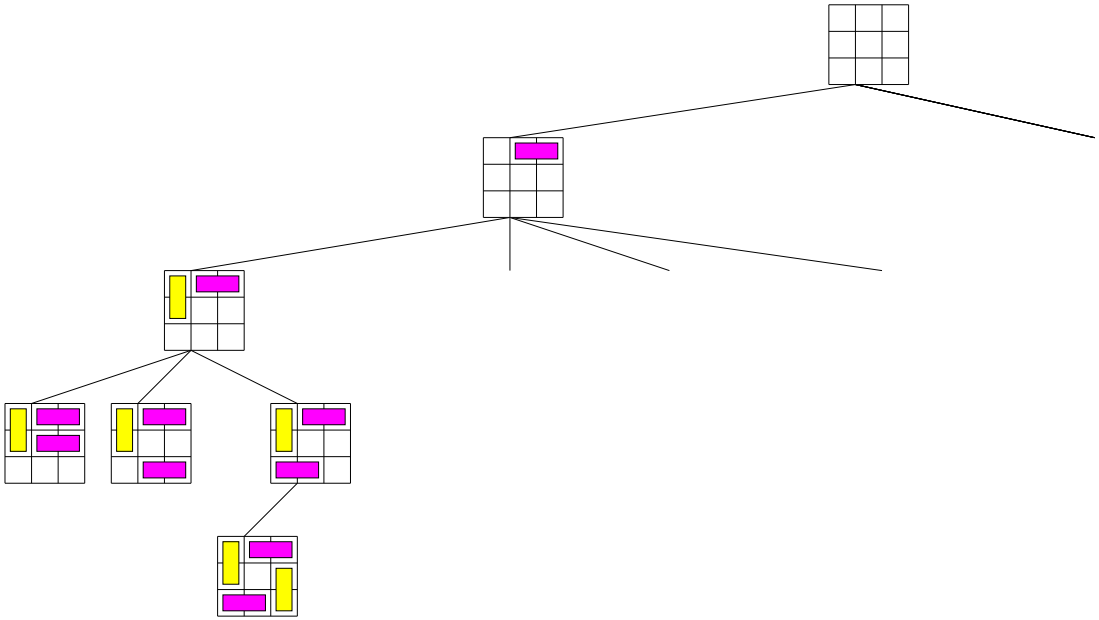
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

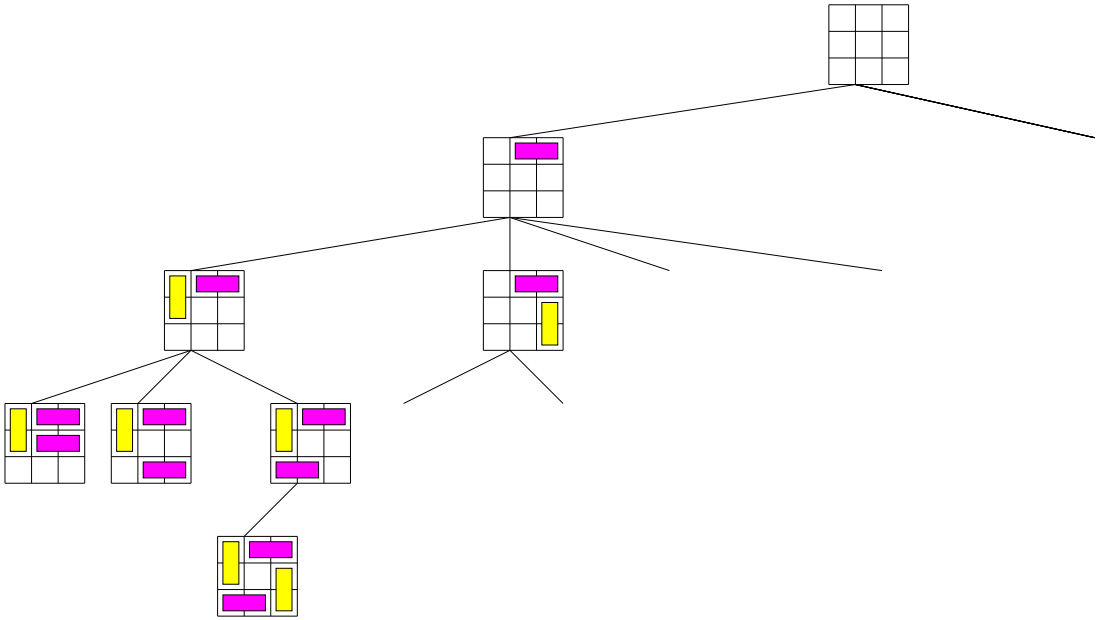
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

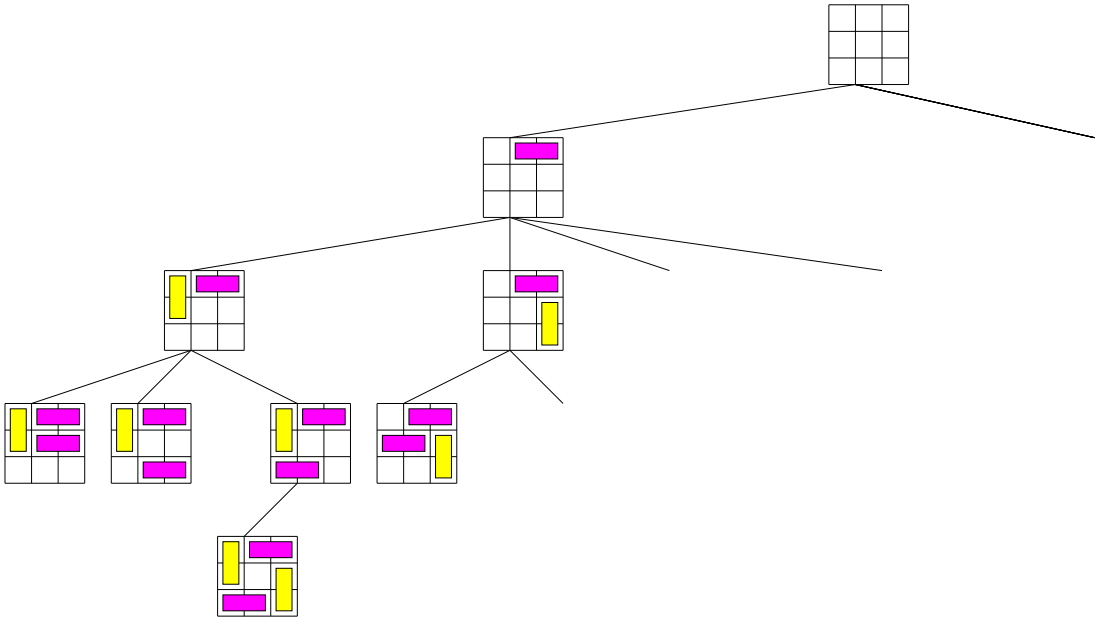
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

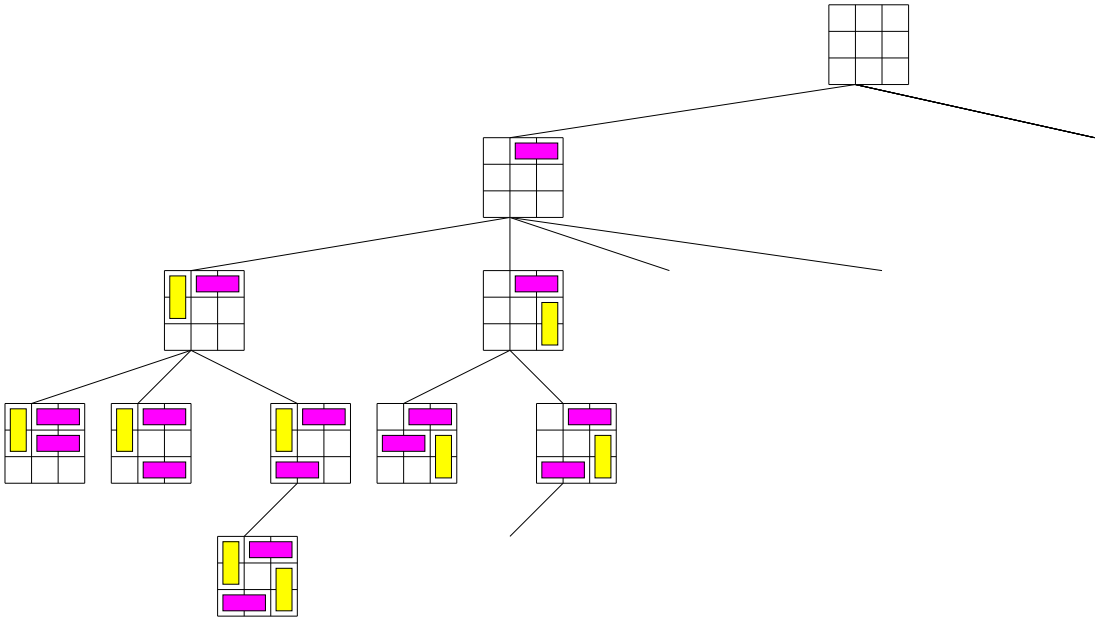
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

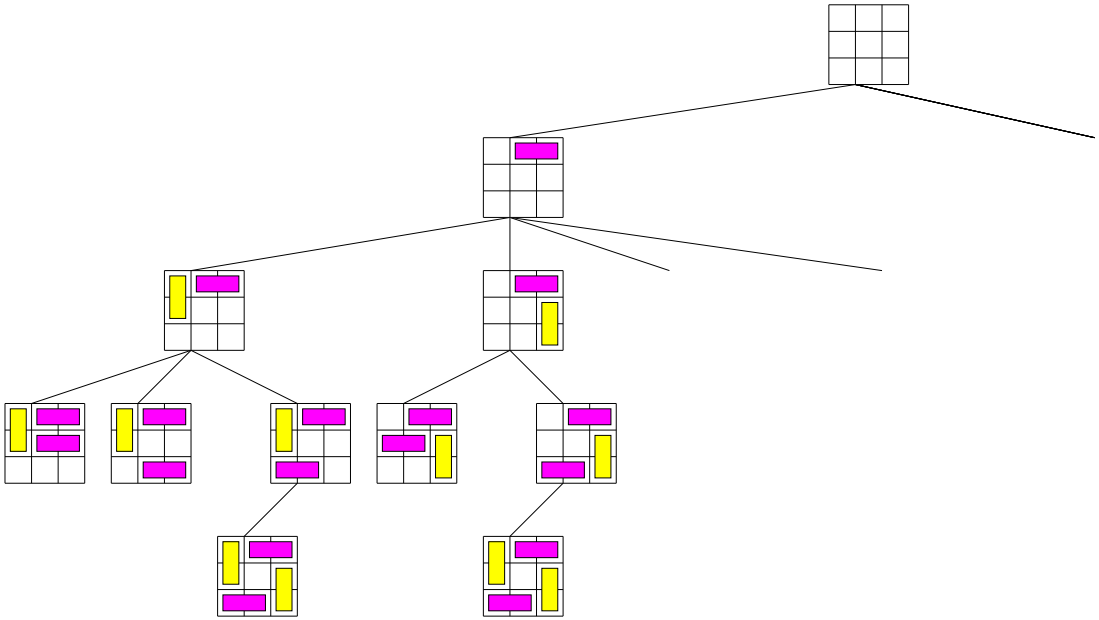
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

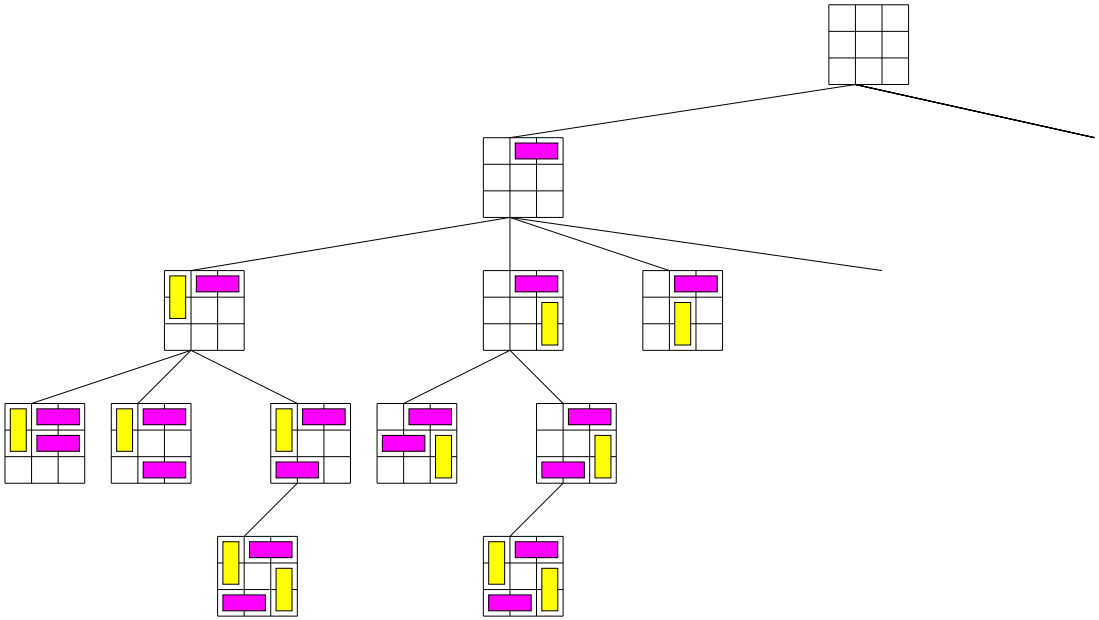
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

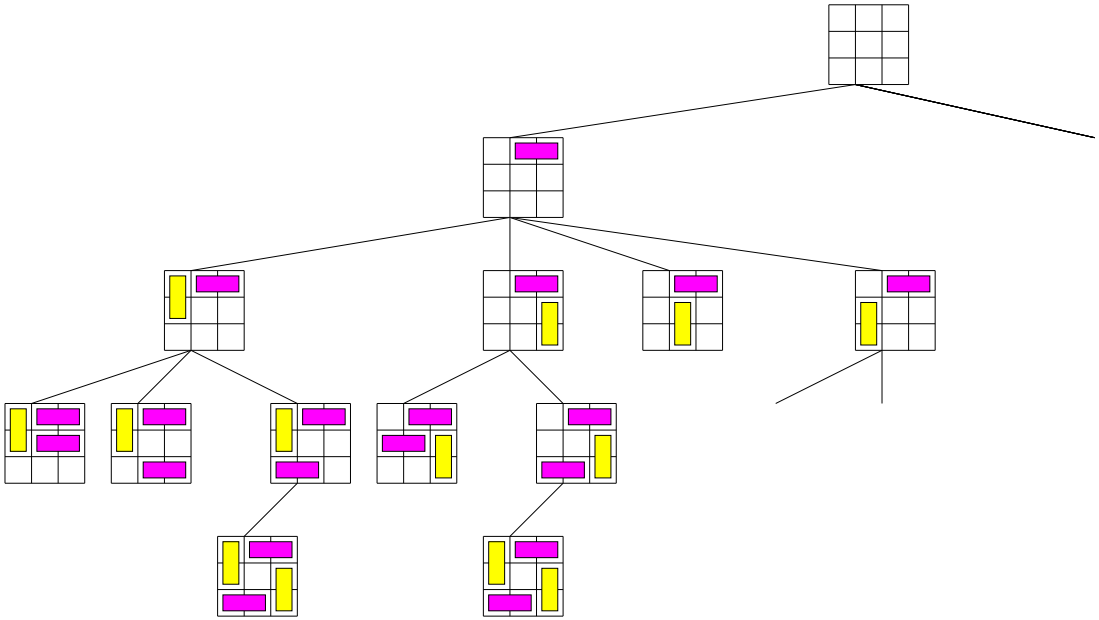
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

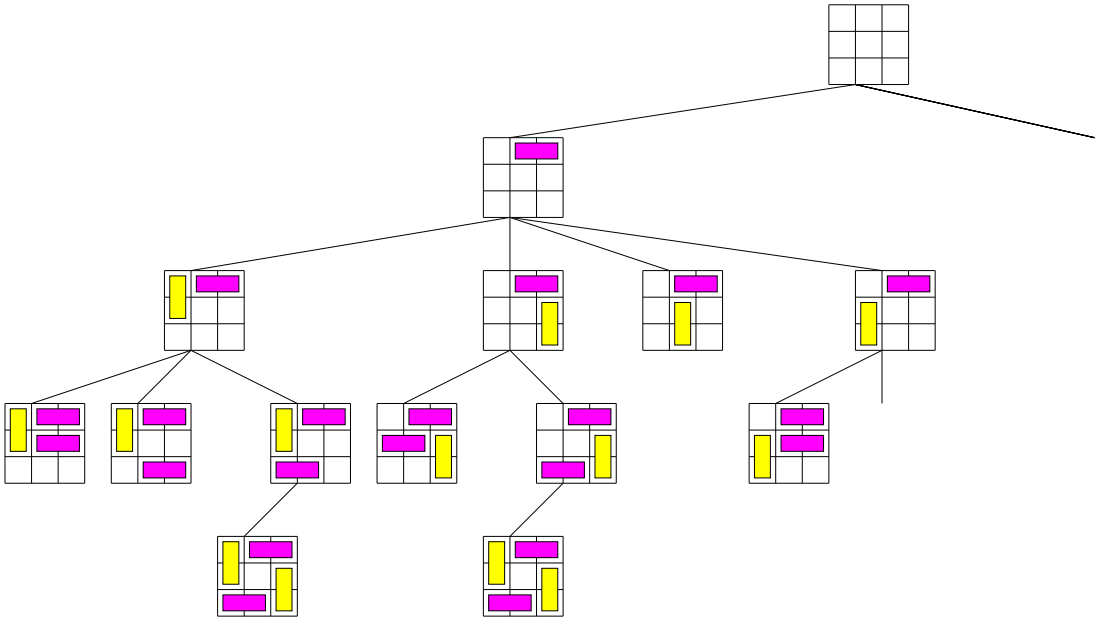
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

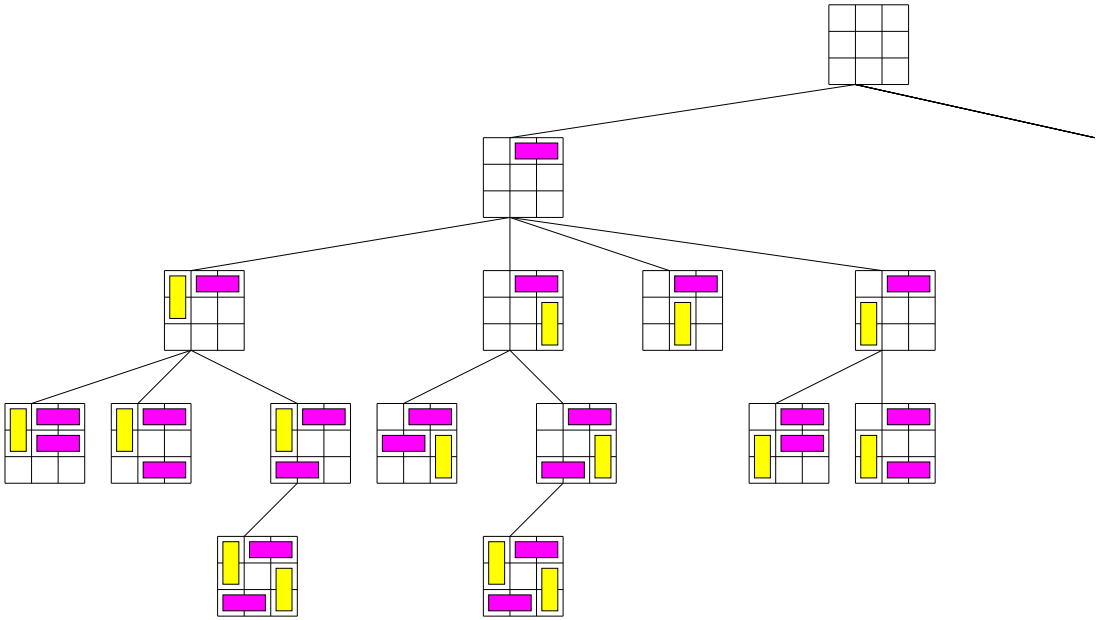
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

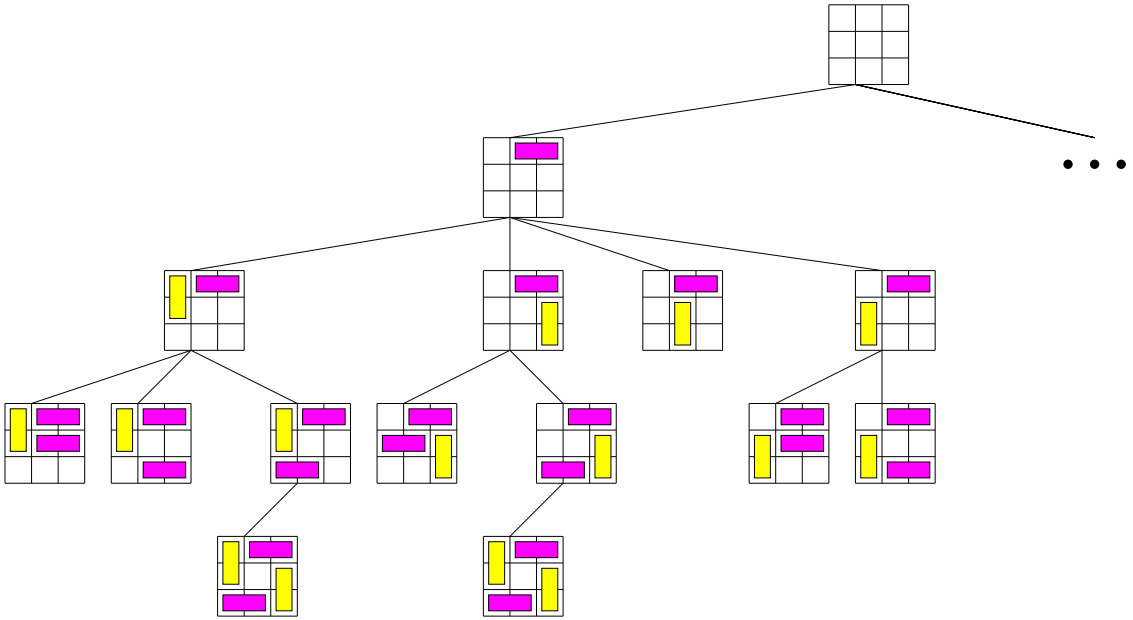
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

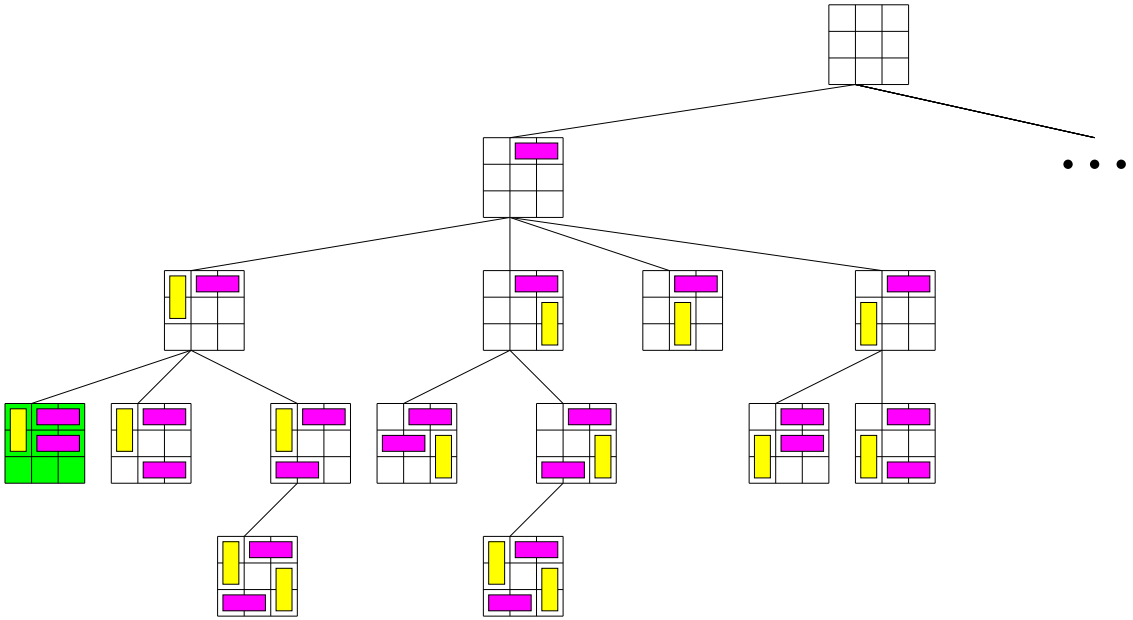
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

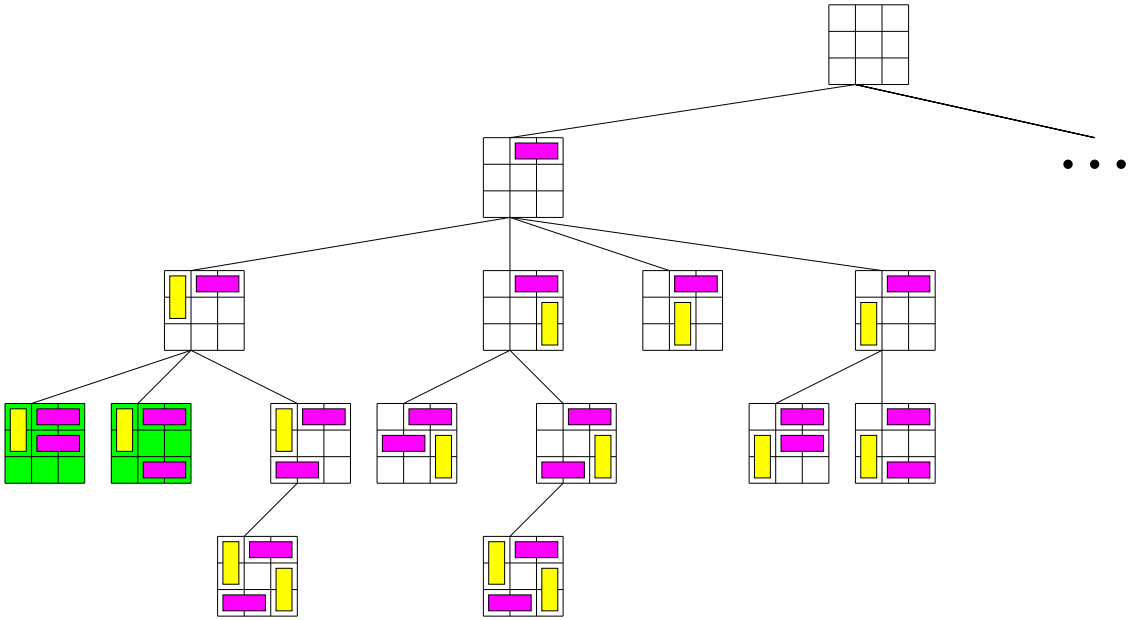
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

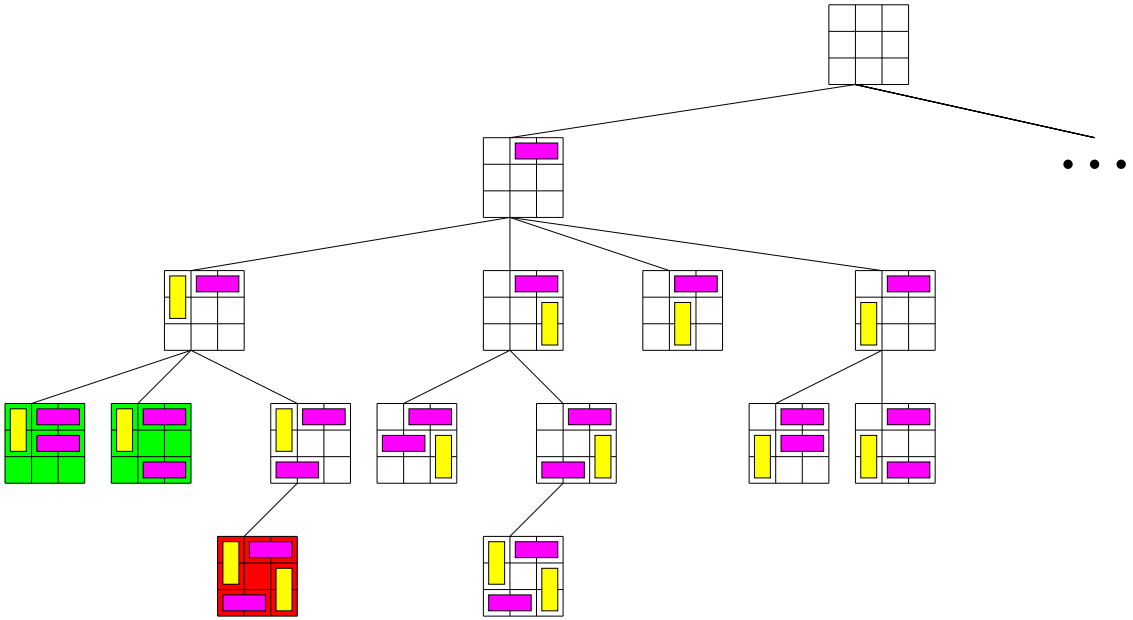
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

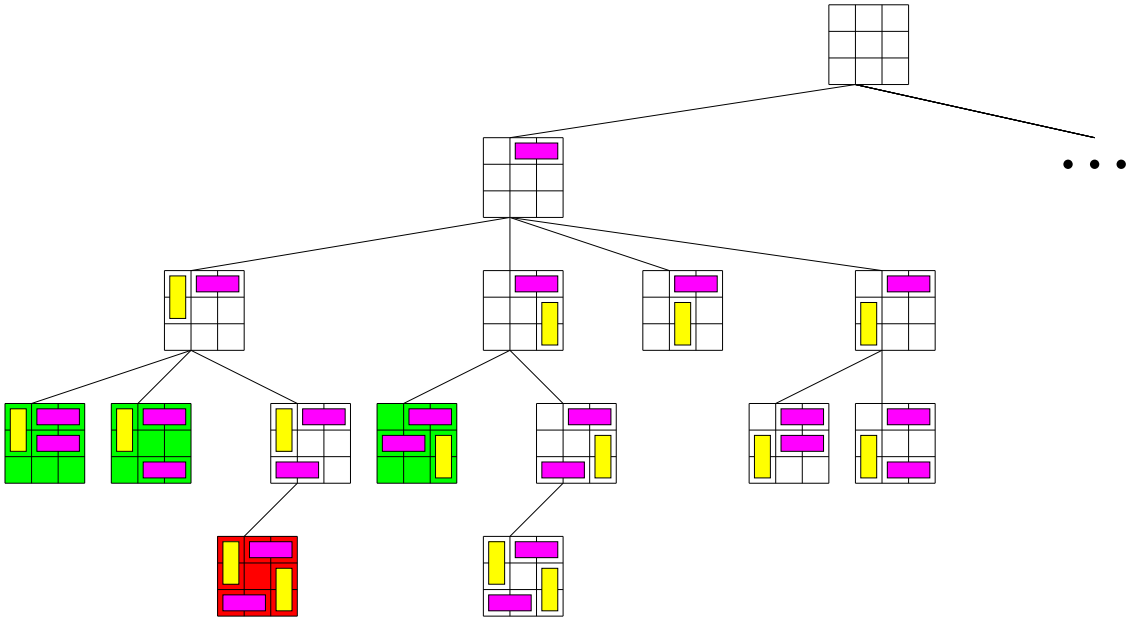
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

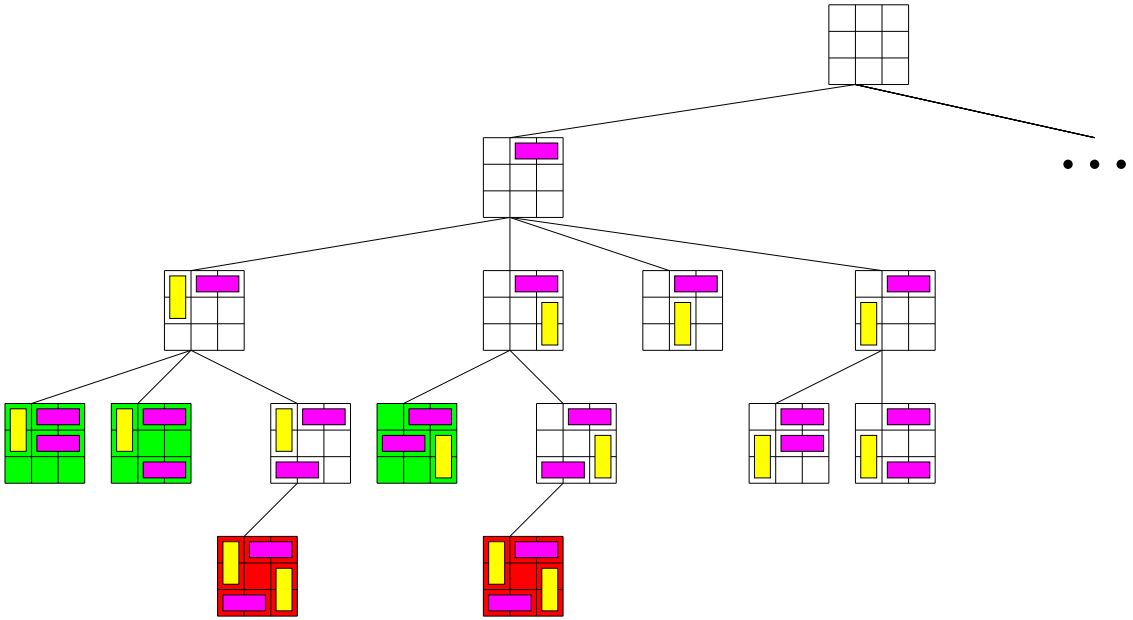
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

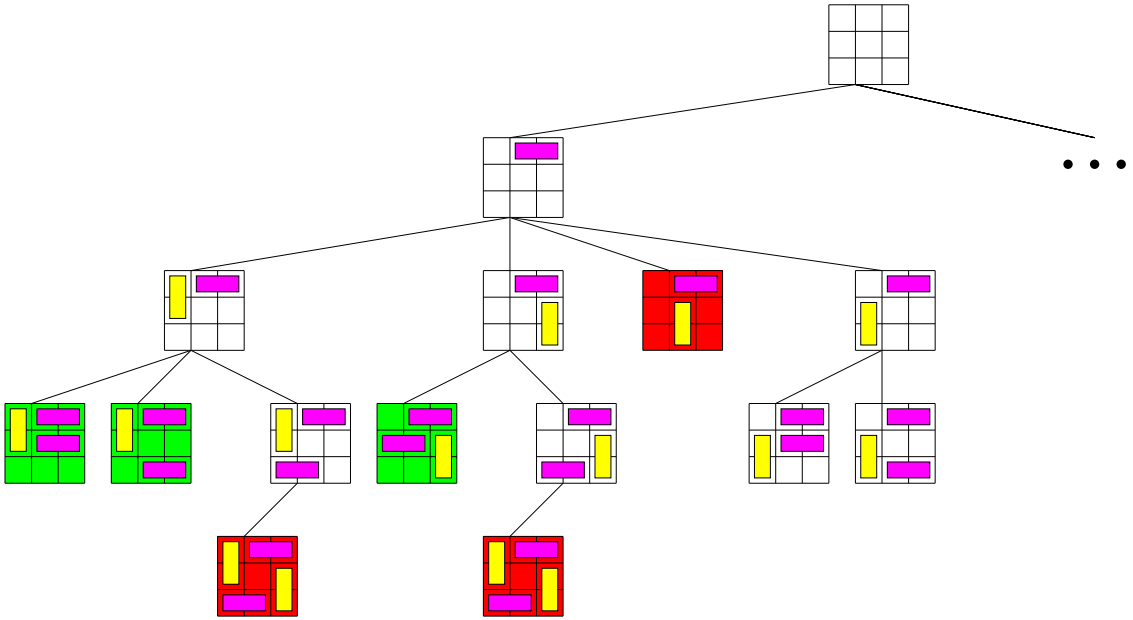
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

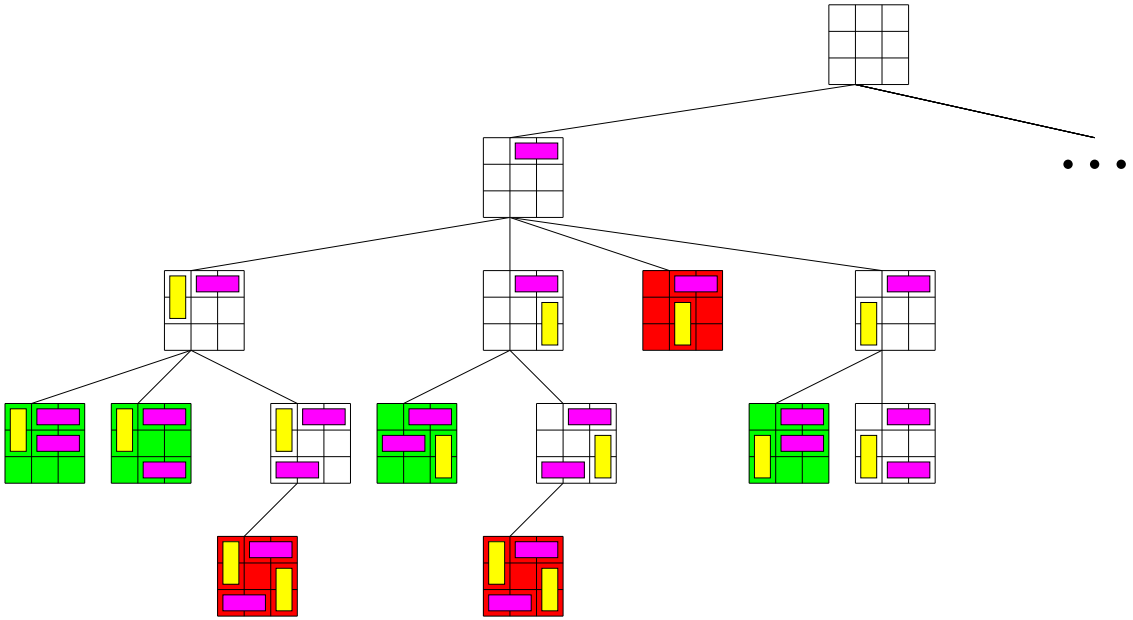
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

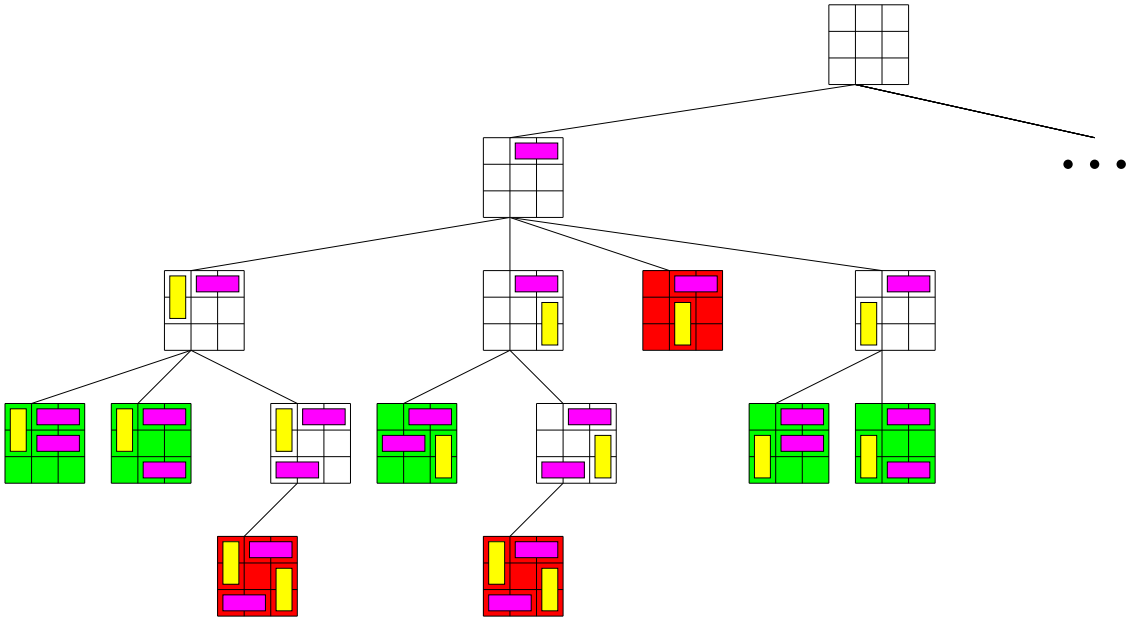
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

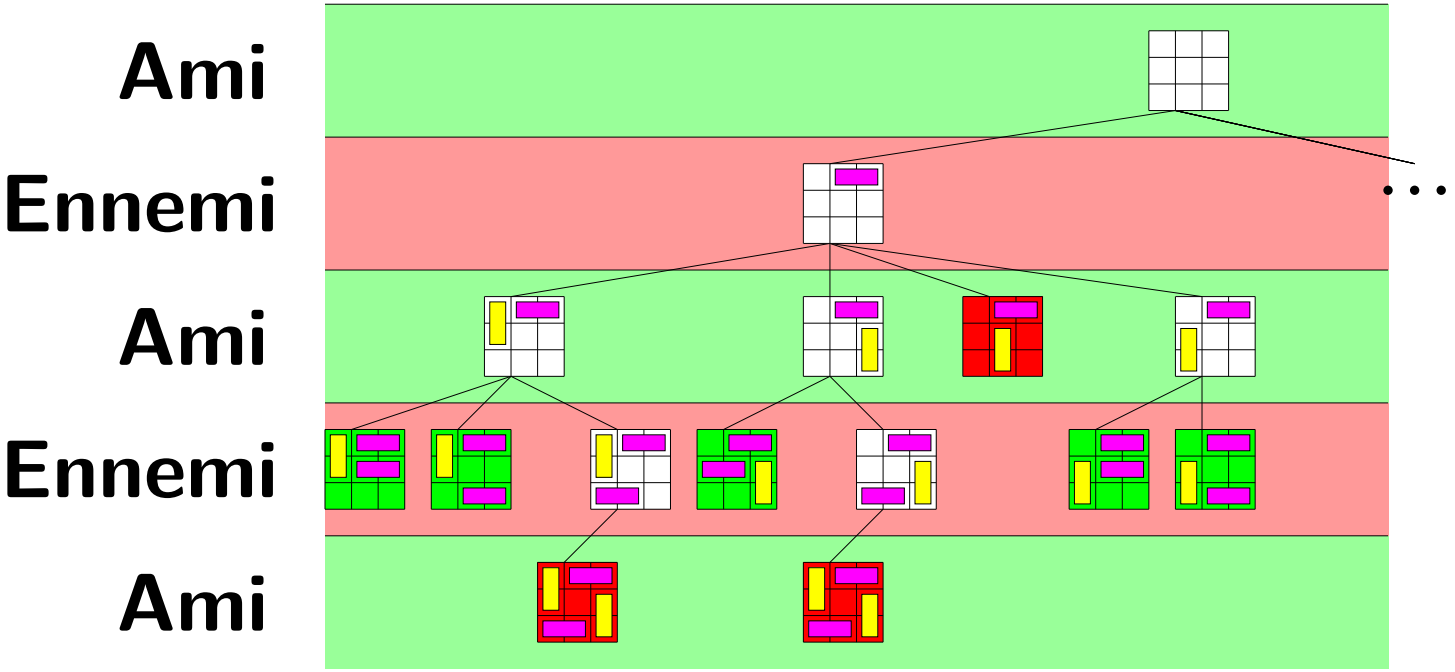
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

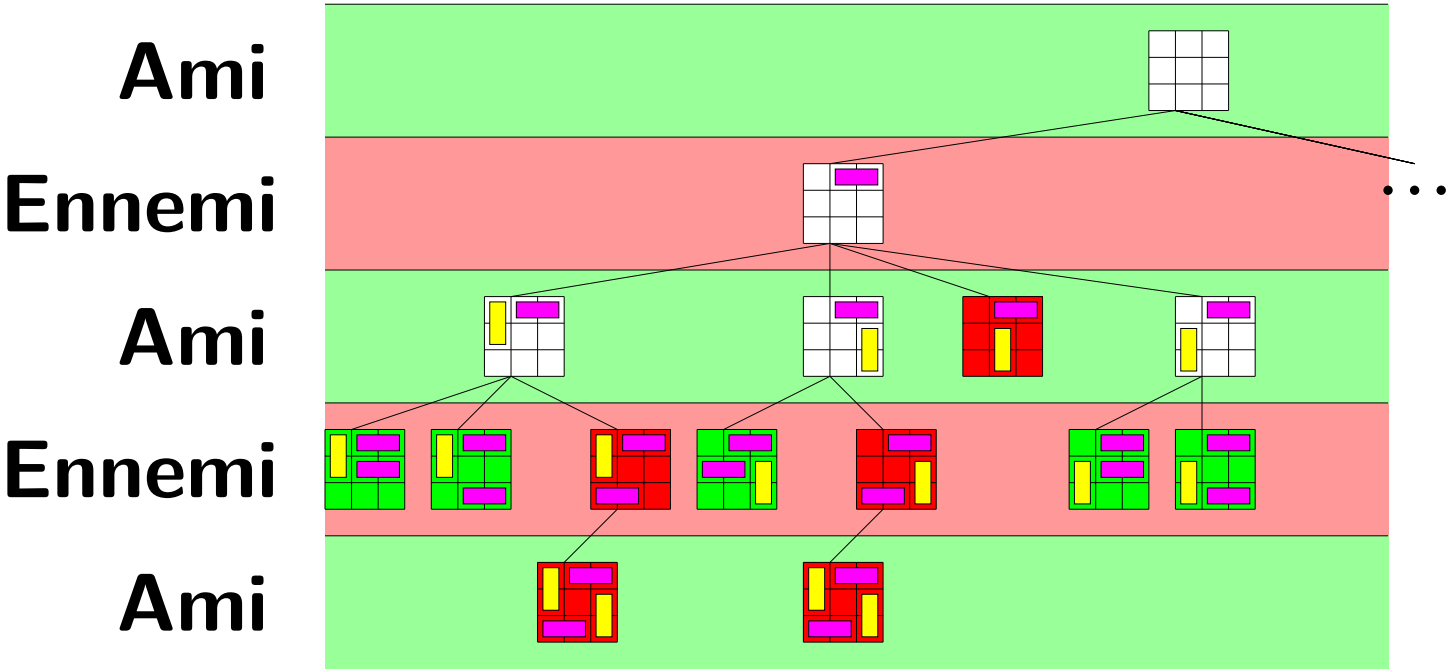
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

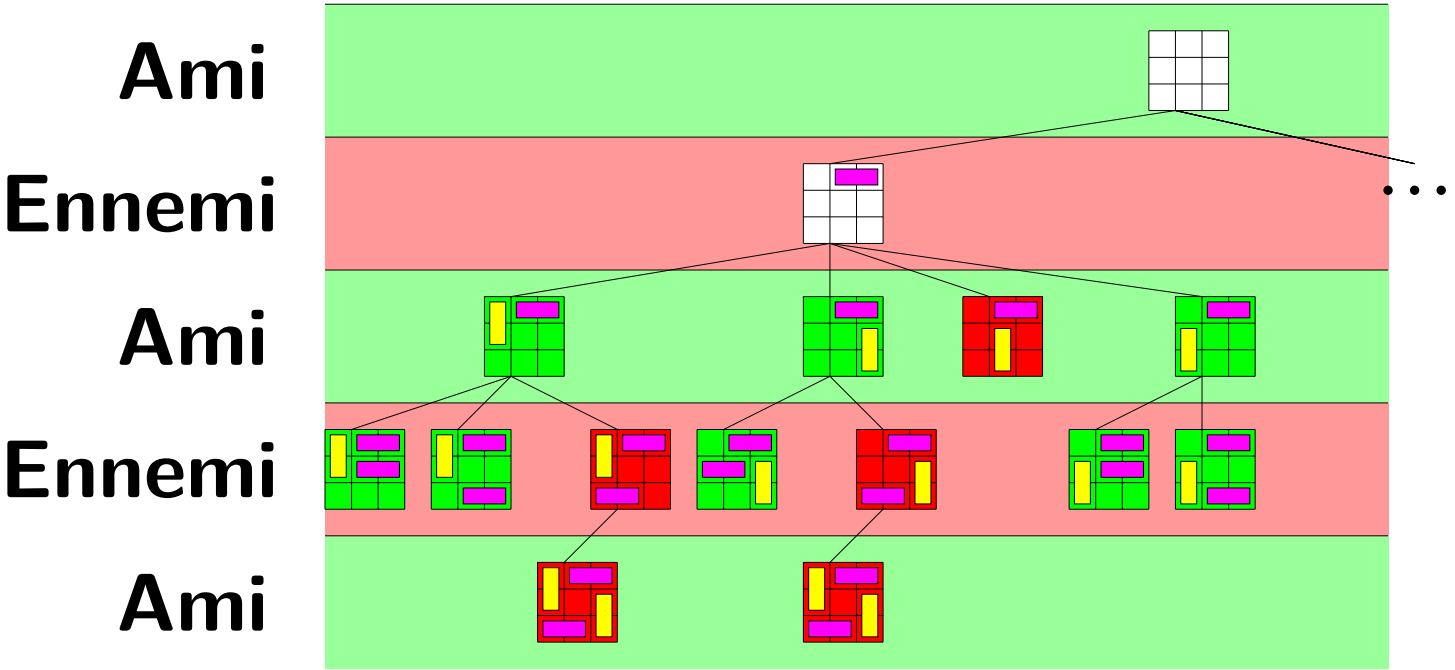
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

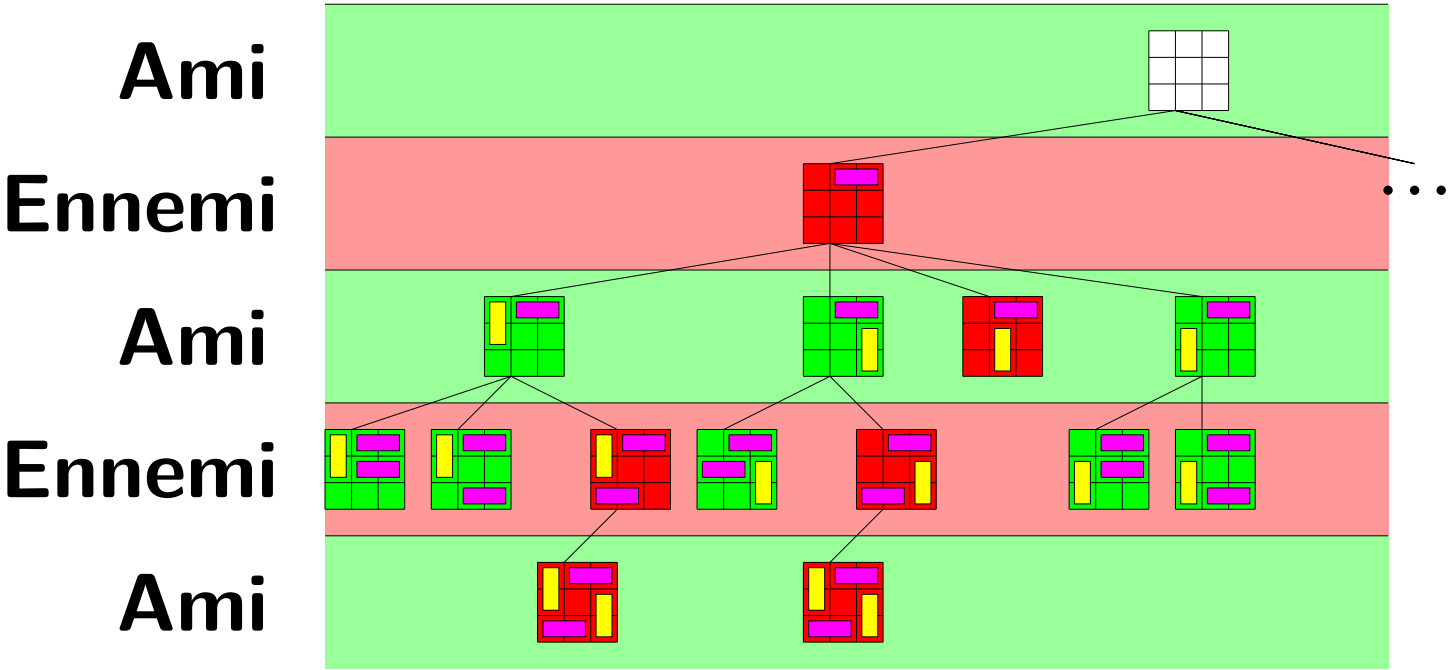
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



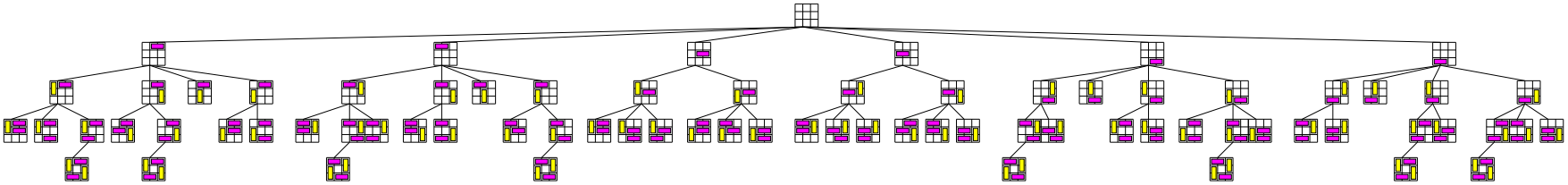
Si Ennemi joue bien, jouer ce coup en premier me fera perdre

Exploration de l'arbre de jeu

Exemple total sur le domino 3 :

Exploration de l'arbre de jeu

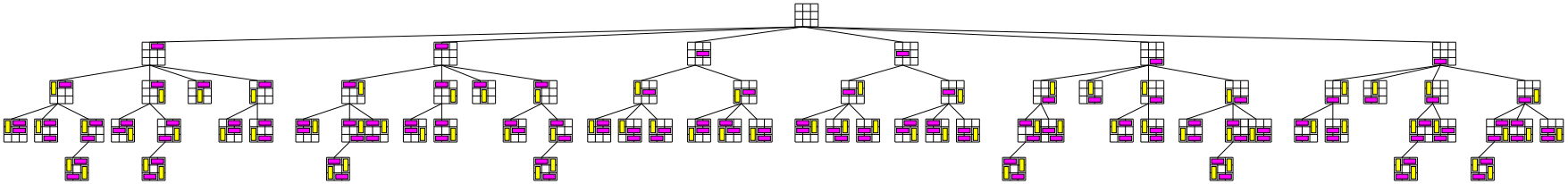
Exemple total sur le domino 3 :



Exploration de tout l'arbre de jeu

Analyse

L'exploration peut se voir comme un parcours classique en profondeur à main gauche dans un arbre décrit *en intention*. Dans l'exemple on a trouvé une **Stratégie Gagnante** en 75 noeuds développés.



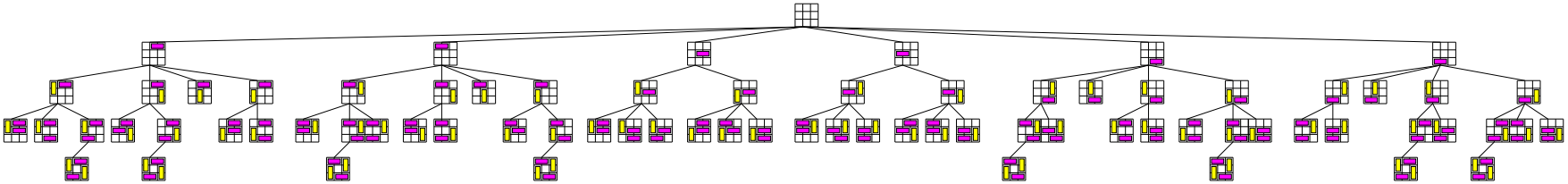
Exploration de tout l'arbre de jeu

Analyse

L'exploration peut se voir comme un parcours classique en profondeur à main gauche dans un arbre décrit *en intention*. Dans l'exemple on a trouvé une **Stratégie Gagnante** en 75 noeuds développés.

Question

Jusqu'où pourrait-on aller avec cette méthode sur un problème simple comme celui-ci ?



Exploration de tout l'arbre de jeu

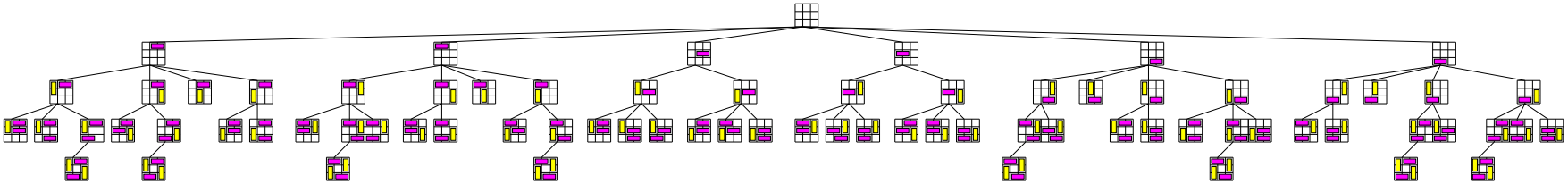
Analyse

L'exploration peut se voir comme un parcours classique en profondeur à main gauche dans un arbre décrit *en intention*. Dans l'exemple on a trouvé une **Stratégie Gagnante** en 75 noeuds développés.

Question

Jusqu'où pourrait-on aller avec cette méthode sur un problème simple comme celui-ci ?

Domino 4, 5, 6, 10, 50, 500 ?



Exploration de tout l'arbre de jeu

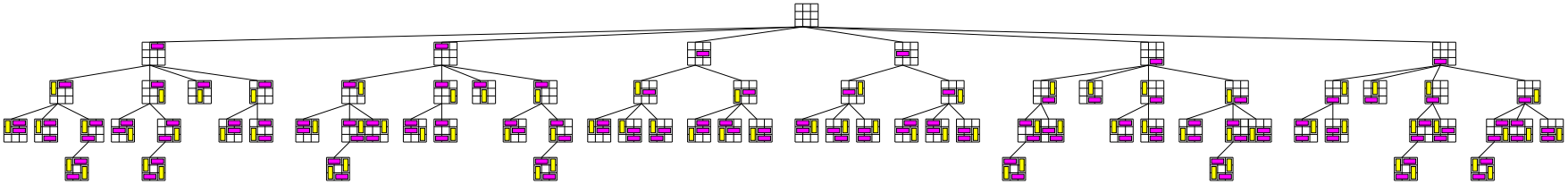
Analyse

L'exploration peut se voir comme un parcours classique en profondeur à main gauche dans un arbre décrit *en intention*. Dans l'exemple on a trouvé une **Stratégie Gagnante** en 75 noeuds développés.

Question

Jusqu'où pourrait-on aller avec cette méthode sur un problème simple comme celui-ci ?

Domino 4, 5, 6, 10, 50, 500 ?
Des idées?



Exploration de tout l'arbre de jeu

Impossibilité, même pour des jeux simples

Quelques expérimentations sur un ordinateur récent, code écrit en C relativement optimisé :

Taille	Nombre de noeuds	Temps
3	75	0.00
4	65 081	0.00
5	2 103 584 600	727.9
6	–	–

Exploration de tout l'arbre de jeu

Impossibilité, même pour des jeux simples

Quelques expérimentations sur un ordinateur récent, code écrit en `C` relativement optimisé :

Taille	Nombre de noeuds	Temps
3	75	0.00
4	65 081	0.00
5	2 103 584 600	727.9
6	–	–

On se trouve face à une **véritable explosion combinatoire**

Exploration de tout l'arbre de jeu

Impossibilité, même pour des jeux simples

Quelques expérimentations sur un ordinateur récent, code écrit en C relativement optimisé :

Taille	Nombre de noeuds	Temps
3	75	0.00
4	65 081	0.00
5	2 103 584 600	727.9
6	–	–

On se trouve face à une **véritable explosion combinatoire**

Comment arriver à aller plus loin ?

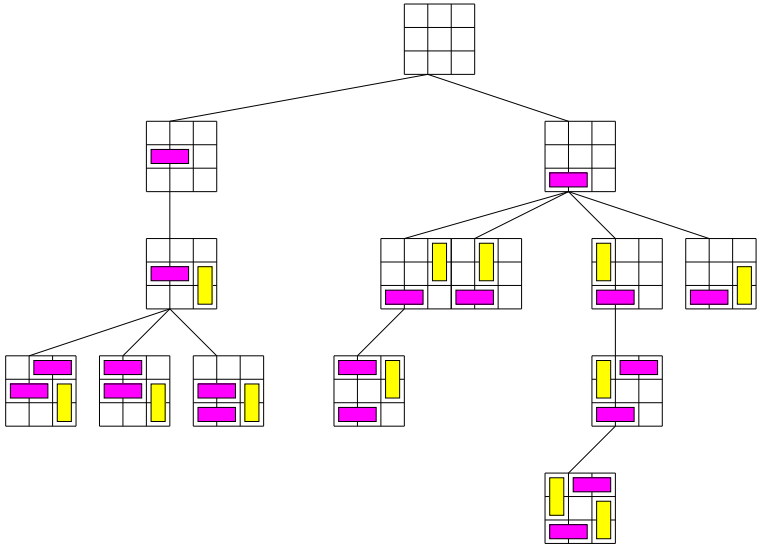
Exploration d'un graphe de jeu

Ne pas réexplorer des sous-graphes commun

Idée : dans certains jeux, un même état peut être atteint par différents endroits. Il faut introduire un mécanisme permettant de ne pas réexplorer des sous-arbres déjà vus.

On peut aller plus loin : considérer tous les états symétriques du plateau de jeu.

L'arbre de jeu précédent se réécrit en



Exploration d'un graphe de jeu

Cela ne change pas le fond des choses

Problèmes : il faut introduire un mécanisme pour retrouver les noeuds déjà explorés. Il faut aussi tous les garder en mémoire.

En pratique :

Taille	Arbre	Graphe	Temps (Graphe)
3	75	23	0.00
4	65 081	2 120	0.00
5	2 103 584 600	718 582	83.3
6	–	–	–

En *profilant* l'exécution, tout le temps est maintenant passé à vérifier si un noeud a déjà été vu.

Toujours pas suffisant ! On ne va pas significativement plus loin !

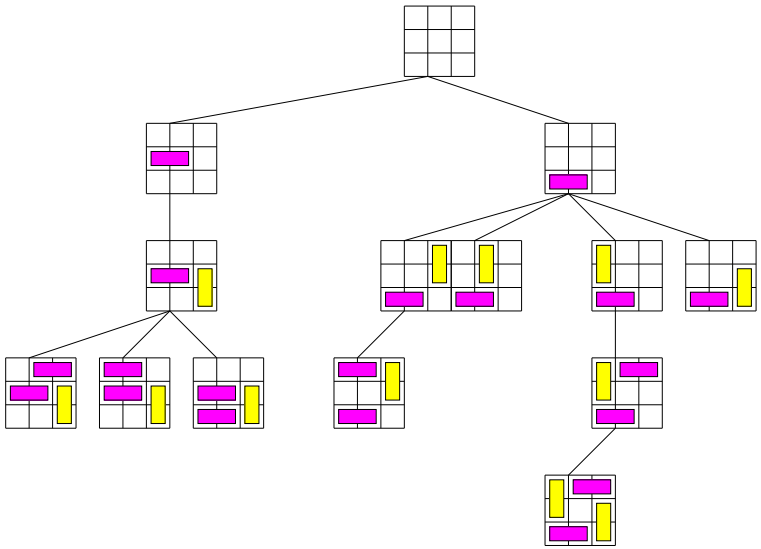
Une première recherche *intelligente*

Les premières *coupes* pas trop idiotes

Idée

Si une branche *amie* mène à la victoire à coups sûrs, ne pas développer les branches voisines.

Si une branche *ennemie* mène à la défaite, ne pas développer les branches voisine.



Une première recherche *intelligente*

Les premières *coupes* pas trop idiotes

- Trouve s'il existe *une stratégie gagnante*
- Permet de couper (élaguer) des parties entières de l'arbre
- Garde les propriétés de l'exploration complète de l'arbre

- Oblige à aller jusqu'aux feuilles de l'arbre (déroulement de toute une partie sur chaque branche non coupée)
- En pratique, aucun jeux ne permet cette technique (ou alors en toute fin de partie)

Toujours pas suffisant ! Il faut pouvoir couper l'arbre sans devoir aller jusqu'aux feuilles

(Petite) Conclusion

Les jeux sont (beaucoup) plus simple que le monde
C'est pour cela qu'on aime y jouer, parce que c'est simple

Dans ce cadre, les ordinateurs peuvent être d'une capacité sur-humaine
Mais cela ne permet pas d'avancer vers une intelligence artificielle générale

Ici, être intelligent, c'est explorer vite (et si possible, bien)

Les approches proposées sont limitées par notre capacité à construire des fonctions heuristiques

Les progrès autour de DeepMind et Alphago utilisent d'autres techniques