

Intelligence Artificielle Jeux de Plateaux (1)

HEY! ALPHAGO! COULD YOU PLEASE STOP
PLAYING GO AND HELP ME WITH THE DINNER?

NO? LAZY AI! JUST GOOD TO PLAY!

LAURENT SIMON
BORDEAUX-INP / LABRI

 @lorensipro
lsimon@labri.fr



Tout (re)commence par des jeux



L'I.A. aime les jeux!

Preuve d'*intelligence*?

- ➊ (Presque) Tous les hommes jouent
- ➋ Certains jeux ont un lourd passé humain
- ➌ Résultats faciles à vulgariser par les chercheurs

Les machines sont adaptées aux univers ludiques

- ➊ Loin des problèmes (physiques) du monde réel
- ➋ Tout est (souvent) formalisable
- ➌ Abstraction complète des problèmes

Robocup, trop de problèmes physiques ?
Dans ce cours on n'étudie *que* les problèmes *abstraits*.

L'I.A. aime les jeux!

Preuve d'*intelligence*?

- (Presque) Tous les hommes jouent
- Certains jeux ont un lourd passé humain
- Résultats faciles à vulgariser par les chercheurs

Les machines sont adaptées aux univers ludiques

- Loin des problèmes (physiques) du monde réel
- Tout est (souvent) formalisable
- Abstraction complète des problèmes

Robocup, trop de problèmes physiques ?
Dans ce cours on n'étudie *que* les problèmes *abstraits*.

L'I.A. aime les jeux!

Preuve d'*intelligence*?

- (Presque) Tous les hommes jouent
- Certains jeux ont un lourd passé humain
- Résultats faciles à vulgariser par les chercheurs

Les machines sont adaptées aux univers ludiques

- ❶ Loin des problèmes (physiques) du monde réel
- ❷ Tout est (souvent) formalisable
- ❸ Abstraction complète des problèmes

Robocup, trop de problèmes physiques ?
Dans ce cours on n'étudie *que* les problèmes *abstraits*.

L'I.A. aime les jeux!

Preuve d'*intelligence*?

- (Presque) Tous les hommes jouent
- Certains jeux ont un lourd passé humain
- Résultats faciles à vulgariser par les chercheurs

Les machines sont adaptées aux univers ludiques

- Loin des problèmes (physiques) du monde réel
- Tout est (souvent) formalisable
- Abstraction complète des problèmes

Robocup, trop de problèmes physiques ?
Dans ce cours on n'étudie *que* les problèmes *abstraits*.

L'I.A. aime les jeux!

Preuve d'*intelligence*?

- (Presque) Tous les hommes jouent
- Certains jeux ont un lourd passé humain
- Résultats faciles à vulgariser par les chercheurs

Les machines sont adaptées aux univers ludiques

- Loin des problèmes (physiques) du monde réel
- Tout est (souvent) formalisable
- Abstraction complète des problèmes

Robocup, trop de problèmes physiques ?
Dans ce cours on n'étudie *que* les problèmes *abstraits*.

L'I.A. aime les jeux!

Preuve d'*intelligence*?

- (Presque) Tous les hommes jouent
- Certains jeux ont un lourd passé humain
- Résultats faciles à vulgariser par les chercheurs

Les machines sont adaptées aux univers ludiques

- Loin des problèmes (physiques) du monde réel
- Tout est (souvent) formalisable
- **Abstraction complète des problèmes**

Robocup, trop de problèmes physiques ?
Dans ce cours on n'étudie *que* les problèmes *abstraits*.

L'I.A. aime les jeux!

Preuve d'*intelligence*?

- (Presque) Tous les hommes jouent
- Certains jeux ont un lourd passé humain
- Résultats faciles à vulgariser par les chercheurs

Les machines sont adaptées aux univers ludiques

- Loin des problèmes (physiques) du monde réel
- Tout est (souvent) formalisable
- Abstraction complète des problèmes



Robocup, trop de problèmes physiques?
Dans ce cours on n'étudie *que* les problèmes
abstraits.

L'I.A. aime les jeux!

Preuve d'*intelligence*?

- (Presque) Tous les hommes jouent
- Certains jeux ont un lourd passé humain
- Résultats faciles à vulgariser par les chercheurs

Les machines sont adaptées aux univers ludiques

- Loin des problèmes (physiques) du monde réel
- Tout est (souvent) formalisable
- Abstraction complète des problèmes



Robocup, trop de problèmes physiques ?
 Dans ce cours on n'étudie que les problèmes
 abstraits.

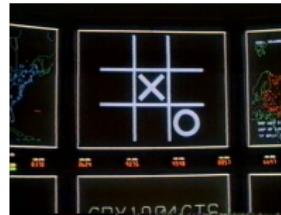
L'I.A. aime les jeux!

Et aussi...

Facile de mesurer expérimentalement les progrès dans un jeu à **deux adversaires**

Vitrine historique de l'I.A.

- ➊ Populaires pour démontrer les nouvelles idées en I.A.
- ➋ Quête Historique de l'I.A. (et dans le futur?)
- ➌ Les résultats sont spectaculaires dans certains jeux
- ➍ Les techniques découvertes doivent servir à d'autres domaines de l'I.A.



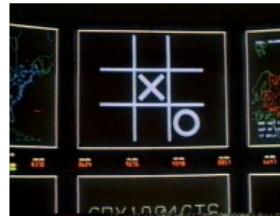
L'I.A. aime les jeux!

Et aussi...

Facile de mesurer expérimentalement les progrès dans un jeu à **deux adversaires**

Vitrine historique de l'I.A.

- Populaires pour démontrer les nouvelles idées en I.A.
- Quête Historique de l'I.A. (et dans le futur ?)
- Les résultats sont spectaculaires dans certains jeux
- Les techniques découvertes doivent servir à d'autres domaines de l'I.A.



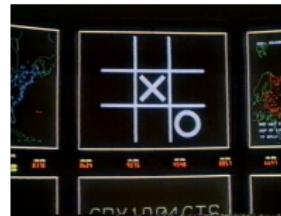
L'I.A. aime les jeux!

Et aussi...

Facile de mesurer expérimentalement les progrès dans un jeu à **deux adversaires**

Vitrine historique de l'I.A.

- Populaires pour démontrer les nouvelles idées en I.A.
- Quête Historique de l'I.A. (et dans le futur?)
- **Les résultats sont spectaculaires dans certains jeux**
- Les techniques découvertes doivent servir à d'autres domaines de l'I.A.



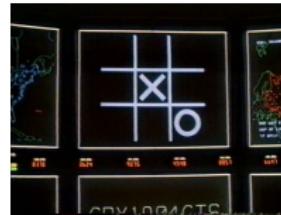
L'I.A. aime les jeux!

Et aussi...

Facile de mesurer expérimentalement les progrès dans un jeu à **deux adversaires**

Vitrine historique de l'I.A.

- Populaires pour démontrer les nouvelles idées en I.A.
- Quête Historique de l'I.A. (et dans le futur?)
- Les résultats sont spectaculaires dans certains jeux
- **Les techniques découvertes doivent servir à d'autres domaines de l'I.A.**



L'I.A. aime les jeux!

Et aussi...

Facile de mesurer expérimentalement les progrès dans un jeu à **deux adversaires**

Vitrine historique de l'I.A.

- Populaires pour démontrer les nouvelles idées en I.A.
- Quête Historique de l'I.A. (et dans le futur?)
- Les résultats sont spectaculaires dans certains jeux
- Les techniques découvertes doivent servir à d'autres domaines de l'I.A.



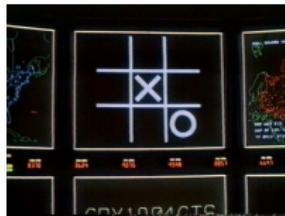
L'I.A. aime les jeux!

Et aussi...

Facile de mesurer expérimentalement les progrès dans un jeu à **deux adversaires**

Vitrine historique de l'I.A.

- Populaires pour démontrer les nouvelles idées en I.A.
- Quête Historique de l'I.A. (et dans le futur?)
- Les résultats sont spectaculaires dans certains jeux
- Les techniques découvertes doivent servir à d'autres domaines de l'I.A.



Un film qui aime l'I.A. qui aime les jeux!

Experience MGM DVD!

★★★★★

An amazingly entertaining thriller.

A masterpiece! — Roger Ebert

Special Features:

- Feature-Length Audio Commentary By Director John Badham And Writers Lawrence Lasker And Walter F. Parkes
- New Dolby Digital 5.1 Soundtrack
- Trivia And Production Notes
- Original Theatrical Trailer

PLUS
SCORE SELECTIONS
SPECIAL FEATURES
LANNINGES

WARGAMES

Jump directly to your favorite scenes!

Scene Selections

1 The Real Father Game	2 Man Who Would Be King
3 The Infiltrator	4 War Games
5 The Infiltrator	6 War Games

SCREEN FORMAT
WIDESCREEN

DOLBY DIGITAL 5.1 SURROUND
English, French and Spanish Language Subtitles

PG **DVD** **NTSC**

Feature Run Time: 1 hour 53 minutes • **COLOR** • 1983 • 907056

© 1983 by the Studio 20 Company, Inc. All Rights Reserved. Licensee: MGM Home Entertainment Inc. All Rights Reserved. Distributed by MGM Home Entertainment, 2500 Broadway, New York, NY 10025. This product contains copyrighted material the unauthorized reproduction, distribution, or exhibition of which may infringe the copyright owner's rights. MGM Home Entertainment, Inc. reserves certain civil and criminal remedies for the unauthorized reproduction, distribution or exhibition of this program. © 1983 by the Studio 20 Company, Inc. All Rights Reserved. Licensee: MGM Home Entertainment, Inc. All Rights Reserved. Distributed by MGM Home Entertainment, 2500 Broadway, New York, NY 10025. This product contains copyrighted material the unauthorized reproduction, distribution or exhibition of which may infringe the copyright owner's rights. MGM Home Entertainment, Inc. reserves certain civil and criminal remedies for the unauthorized reproduction, distribution or exhibition of this program.

All MGM/D20 titles are presented with digital and high definition television and DVD video encoded DVD-Audio tracks.

MGM

Matthew Broderick DABNEY COLEMAN JOHN WOOD ALLY SHEEDY

WARGAMES

MISSILE WARNING

WARGAMES

A man (Matthew Broderick) plays a computer game that allows him to control the world's nuclear missiles. He becomes so engrossed in the game that he accidentally launches one of the missiles.

Les jeux aiment l'I.A.

Les jeux vidéos

« **On achète un jeu pour ses qualités graphiques mais on reste dessus pour ses qualités de jeux.** »

De l'argent à profusion ?

Les éditeurs de jeux font de plus en plus d'effort au niveau de l'I.A. La puissance de calcul des consoles (PS5, XOne, Nintendo Switch) est en partie dédiées au calcul pur, donc aux I.A. des jeux.

Mais, comme pour les robots, c'est un *autre problème*

Les jeux aiment l'I.A.

Les jeux vidéos

« On achète un jeu pour ses qualités graphiques, mais on reste dessus pour ses qualités de jeux.»

De l'argent à profusion ?

Les éditeurs de jeux font de plus en plus d'effort au niveau de l'I.A. La puissance de calcul des consoles (PS5, XOne, Nintendo Switch) est en partie dédiées au calcul pur, donc aux I.A. des jeux.

Mais, comme pour les robots, c'est un *autre problème*

Contrainte temporelle forte

Les jeux aiment l'I.A.

Les jeux vidéos

« On achète un jeu pour ses qualités graphiques, mais on reste dessus pour ses qualités de jeux.»

De l'argent à profusion ?

Les éditeurs de jeux font de plus en plus d'effort au niveau de l'I.A. La puissance de calcul des consoles (PS5, XOne, Nintendo Switch) est en partie dédiées au calcul pur, donc aux I.A. des jeux.

Mais, comme pour les robots, c'est un *autre problème*

- ➊ Contrainte temps-réel forte
- ➋ Triche pour rendre les I.A. réalistes
- ➌ ...

Les jeux aiment l'I.A.

Les jeux vidéos

« On achète un jeu pour ses qualités graphiques, mais on reste dessus pour ses qualités de jeux.»

De l'argent à profusion ?

Les éditeurs de jeux font de plus en plus d'effort au niveau de l'I.A. La puissance de calcul des consoles (PS5, XOne, Nintendo Switch) est en partie dédiées au calcul pur, donc aux I.A. des jeux.

Mais, comme pour les robots, c'est un *autre problème*

- ➊ Contrainte temps-réel forte
- ➋ Triche pour rendre les I.A. réalistes
- ➌ ...

Les jeux aiment l'I.A.

Les jeux vidéos

« On achète un jeu pour ses qualités graphiques, mais on reste dessus pour ses qualités de jeux.»

De l'argent à profusion ?

Les éditeurs de jeux font de plus en plus d'effort au niveau de l'I.A. La puissance de calcul des consoles (PS5, XOne, Nintendo Switch) est en partie dédiées au calcul pur, donc aux I.A. des jeux.

Mais, comme pour les robots, c'est un *autre problème*

- Contrainte temps-réel forte
- Tricherie pour rendre les I.A. *réalistes*



Les jeux aiment l'I.A.

Les jeux vidéos

« On achète un jeu pour ses qualités graphiques, mais on reste dessus pour ses qualités de jeux.»

De l'argent à profusion ?

Les éditeurs de jeux font de plus en plus d'effort au niveau de l'I.A. La puissance de calcul des consoles (PS5, XOne, Nintendo Switch) est en partie dédiées au calcul pur, donc aux I.A. des jeux.

Mais, comme pour les robots, c'est un *autre problème*

- Contrainte temps-réel forte
- Tricherie pour rendre les I.A. réalistes
- ...

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- ➊ Règles de jeu formalisables et précises
- ➋ Deux joueurs adversaires
- ➌ Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
- ➍ Jeux ouverts

Les résultats pourront être étendus aux jeux :

- ➎ À plusieurs joueurs (mais toujours à somme zéro)
- ➏ Avec tirage de dés (ou de lettres)
- ➐ Avec contrainte temps réel (Ah?)
- ➑ Avec connaissance imparfaite (Ah Ah?)

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- ➊ Règles de jeu formalisables et précises
- ➋ Deux joueurs adversaires
 - ➌ Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
 - ➍ Jeux ouverts

Les résultats pourront être étendus aux jeux :

- ➎ À plusieurs joueurs (mais toujours à somme zéro)
- ➏ Avec tirage de dés (ou de lettres)
- ➐ Avec contrainte temps réel (Ah?)
- ➑ Avec connaissance imparfaite (Ah Ah?)

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- ➊ Règles de jeu formalisables et précises
- ➋ Deux joueurs adversaires
- ➌ Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
- ➍ Jeux ouverts

Les résultats pourront être étendus aux jeux :

- ➎ À plusieurs joueurs (mais toujours à somme zéro)
- ➏ Avec tirage de dés (ou de lettres)
- ➐ Avec contrainte temps réel (Ah?)
- ➑ Avec connaissance imparfaite (Ah Ah?)

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- Règles de jeu formalisables et précises
- Deux joueurs adversaires
- Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
- Jeux ouverts

Les résultats pourront être étendus aux jeux :

- À plusieurs joueurs (mais toujours à somme zéro)
- Avec tirage de dés (ou de lettres)
- Avec contrainte temps réel (Ah?)
- Avec connaissance imparfaite (Ah Ah?)

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- Règles de jeu formalisables et précises
- Deux joueurs adversaires
- Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
- Jeux ouverts

Les résultats pourront être étendus aux jeux :

- À plusieurs joueurs (mais toujours à somme zéro)
- Avec tirage de dés (ou de lettres)
- Avec contrainte temps réel (Ah?)
- Avec connaissance imparfaite (Ah Ah?)

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- Règles de jeu formalisables et précises
- Deux joueurs adversaires
- Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
- Jeux ouverts

Les résultats pourront être étendus aux jeux :

- À plusieurs joueurs (mais toujours à somme zéro)
- Avec tirage de dés (ou de lettres)
- Avec contrainte temps réel (Ah?)
- Avec connaissance imparfaite (Ah Ah?)

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- Règles de jeu formalisables et précises
- Deux joueurs adversaires
- Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
- Jeux ouverts

Les résultats pourront être étendus aux jeux :

- À plusieurs joueurs (mais toujours à somme zéro)
- Avec tirage de dés (ou de lettres)
- Avec contrainte temps réel (Ah?)
- Avec connaissance imparfaite (Ah Ah?)

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- Règles de jeu formalisables et précises
- Deux joueurs adversaires
- Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
- Jeux ouverts

Les résultats pourront être étendus aux jeux :

- À plusieurs joueurs (mais toujours à somme zéro)
- Avec tirage de dés (ou de lettres)
- Avec contrainte temps réel (Ah?)
- Avec connaissance imparfaite (Ah Ah?)

Les jeux que l'on aime dans ce cours

On ne va pas étudier les jeux physiques ou avec contraintes temps réel forte.

Ce que l'on va étudier

- Règles de jeu formalisables et précises
- Deux joueurs adversaires
- Jeux à somme zéro (ce que l'un perd, l'autre le gagne)
- Jeux ouverts

Les résultats pourront être étendus aux jeux :

- À plusieurs joueurs (mais toujours à somme zéro)
- Avec tirage de dés (ou de lettres)
- Avec contrainte temps réel (Ah?)
- Avec connaissance imparfaite (Ah Ah?)

Historique des jeux

Article fondateur de l'I.A.

Philosophical Magazine, Ser.7, Vol. 41, No. 314 - March 1950.

XXII. Programming a Computer for Playing Chess¹

By CLAUDE E. SHANNON

Bell Telephone Laboratories, Inc., Murray Hill, N.J.²

[Received November 8, 1949]

Le premier article d'I.A. est un article sur les jeux (Shannon, 1950).

Philosophical Magazine, Ser.7, Vol. 41, No. 314 - March 1950.

XXII. Programming a Computer for Playing Chess¹

By CLAUDE E. SHANNON

Bell Telephone Laboratories, Inc., Murray Hill, N.J.²

[Received November 8, 1949]

1. INTRODUCTION

This paper is concerned with the problem of constructing a computing routine or "program" for a modern general-purpose computer which will enable it to play chess. Although perhaps of no practical importance, the question is of theoretical interest, and it is hoped that a satisfactory solution of this problem will act as a wedge in attacking other problems of a similar nature and of greater significance. Some possibilities in this direction are:

- (1)Machines for designing filters, equalizers, etc.
- (2)Machines for designing relay and switching circuits.
- (3)Machines which will handle routing of telephone calls based on the individual circumstances rather than by fixed patterns.
- (4)Machines for performing symbolic (non-numerical) mathematical operations.
- (5)Machines capable of translating from one language to another.
- (6)Machines for making strategic decisions in simplified military operations.
- (7)Machines capable of orchestrating a melody.
- (8)Machines capable of logical deduction.

It is believed that all of these and many other devices of a similar nature are possible developments in the immediate future. The techniques developed for modern electronic and relay type computers make them not only theoretical possibilities, but in several cases worthy of serious consideration from the economic point of view.

Machines of the general type are an extension over the ordinary use of numerical computers in several ways. First, the entities dealt with are not primarily numbers, but rather symbols, pointers, circuit conditions, expressed words, etc. Second, the proper procedures involve general principles, something of the nature of a search, involving considerable trial and error, rather than a strict, unshakable computing process. Finally, the solutions of these problems are not merely right or wrong but have a continuous range of "quality" from the best down to the worst. We might be satisfied with a machine that designed good filters even though they were not always the best possible.

¹ First presented at the National IRE Convention, March 9, 1949, New York, U.S.A.

² Communicated by the Author

Historique des jeux

Article fondateur de l'I.A.

1. INTRODUCTION

This paper is concerned with the problem of constructing a computing routine or "program" for a modern general purpose computer which will enable it to play chess. Although perhaps of no practical importance, the question is of theoretical interest, and it is hoped that a satisfactory solution of this problem will act as a wedge in attacking other problems of a similar nature and of greater significance. Some possibilities in this direction are: -

- (1)Machines for designing filters, equalizers, etc.
- (2)Machines for designing relay and switching circuits.
- (3)Machines which will handle routing of telephone calls based on the individual circumstances rather than by fixed patterns.
- (4)Machines for performing symbolic (non-numerical) mathematical operations.
- (5)Machines capable of translating from one language to another.
- (6)Machines for making strategic decisions in simplified military operations.
- (7)Machines capable of orchestrating a melody.
- (8)Machines capable of logical deduction.

Historique des jeux

Article fondateur de l'I.A.

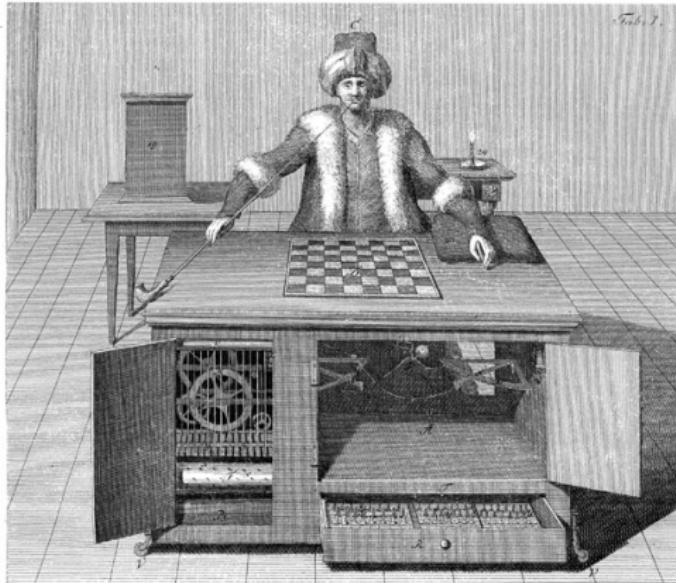
1. INTRODUCTION

This paper is concerned with the problem of constructing a computing routine or "program" for a modern general purpose computer which will enable it to play chess. Although perhaps of no practical importance, the question is of theoretical interest, and it is hoped that a satisfactory solution of this problem will act as a wedge in attacking other problems of a similar nature and of greater significance. Some possibilities in this direction are: -

- (1)Machines for designing filters, equalizers, etc.
- (2)Machines for designing relay and switching circuits.
- (3)Machines which will handle routing of telephone calls based on the individual circumstances rather than by fixed patterns.
- (4)Machines for performing symbolic (non-numerical) mathematical operations.
- (5)Machines capable of translating from one language to another.
- (6)Machines for making strategic decisions in simplified military operations.
- (7)Machines capable of orchestrating a melody.
- (8)Machines capable of logical deduction.

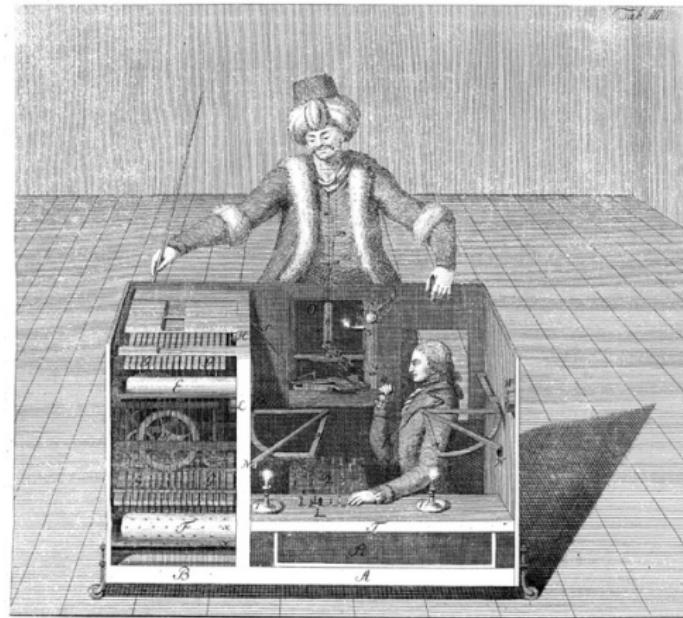
Historique des jeux

Une machine qui joue aux Échecs en 1769 ?



Historique des jeux

Une machine qui joue aux Échecs en 1769 ?



Historique des jeux

Une machine d'aujourd'hui qui joue aux Échecs comme le Turc ?

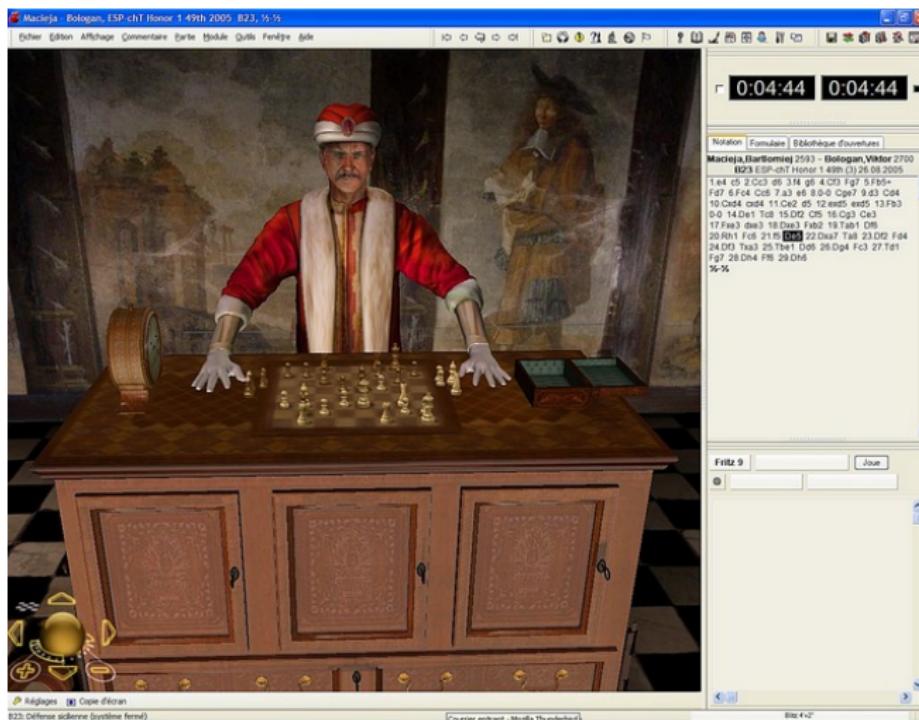


Image du *Turc mécanique* issue de Fritz 9 (2006).

Historique des jeux

On commence par des Échecs ?

1945

- 1945 Turing présente les échecs comme ce que peuvent faire les ordinateurs
- 1946 Turing parle d'intelligence des machines, en relation avec les échecs
- 1950 Turing écrit le premier programme de jeux (la même année que le test de Turing)
- 1950 Shannon écrit le premier article scientifique sur les échecs
- 1951 Turing simule le premier programme d'échecs à la main contre un joueur assez mauvais. Il perd.
- 1952 A. Glennie est le premier humain à battre un ordinateur aux échecs (*TurboChamp*, de Turing)
- 1957 H. Simon prédit que d'ici 10 ans, le champion du monde d'échecs sera un ordinateur.

1957

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- ➊ A. Samuel étudie les dames dès 1959
- ➋ En 1963, son programme gagne contre un champion
- ➌ En 1970, il est battu par un programme de la Duke University dans un match court
- ➍ Ils font partie des 10 meilleurs joueurs mondiaux
- ➎ En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive.
- ➏ Chinook gagne contre un champion mondial en 1992.
- ➐ Après avoir perdu en 1992, *Chinook* gagne en 1994.
- ➑ *Chinook* n'a jamais plus perdu un match.
- ➒ Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, **les Dames sont résolues.** (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive.
- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, **les Dames sont résolues.** (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive.

- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, les Dames sont résolues. (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive. *The World Man-Machine Championship* est créée.
- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, les Dames sont résolues. (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive. *The World Man-Machine Championship* est créée
- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, les Dames sont résolues. (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive. *The World Man-Machine Championship* est créée
- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, les Dames sont résolues. (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive. *The World Man-Machine Championship* est créée
- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, les Dames sont résolues. (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive. *The World Man-Machine Championship* est créée
- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, les Dames sont résolues. (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive. *The World Man-Machine Championship* est créée
- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, les Dames sont résolues. (voir site web de Schaeffer sur Chinook)

Historique des jeux

Les premiers succès : Les dames

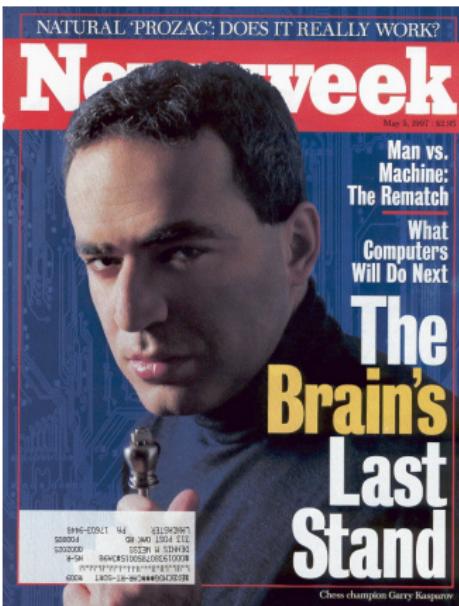
Premier jeu à battre un champion du monde humain

- A. Samuel étudie les dames dès 1959
- En 1963, son programme gagne contre un champion
- En 1970, il est battu par un programme de la Duke University dans un match court
- Ils font partie des 10 meilleurs joueurs mondiaux
- En 1989, des intérêts commerciaux entrent en jeu et *Chinook* [Schaeffer 1989] arrive. *The World Man-Machine Championship* est créée
- Après avoir perdu en 1992, *Chinook* gagne en 1994.
- *Chinook* n'a jamais plus perdu un match.
- Il a été retiré des compétitions humaines en 1997.

Aujourd'hui, **les Dames sont résolues.** (voir site web de Schaeffer sur Chinook)



1997 : Kasparov VS Deep(er) Blue



2016 : Lee Sedol vs AlphaGo



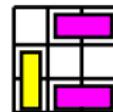
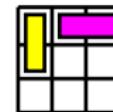
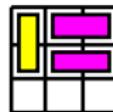
recherche totale

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 .
Noir joue horizontalement et blanc verticalement.



Un jeu simple mais illustratif

Nous allons utiliser ce jeu simple pour schématiser les approches

Exploration de l'arbre de jeu

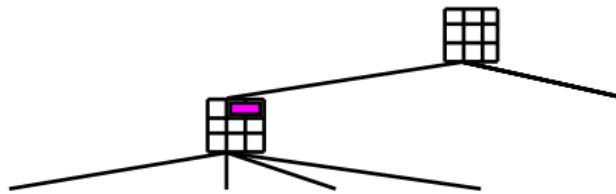
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

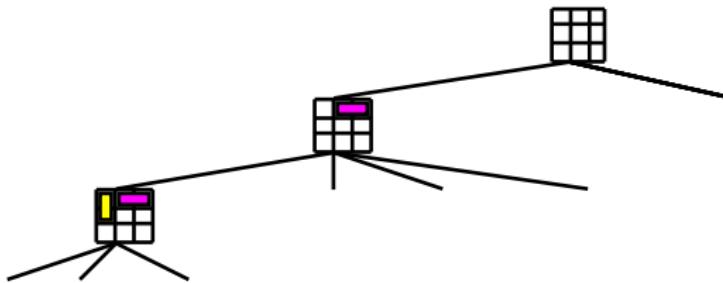
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

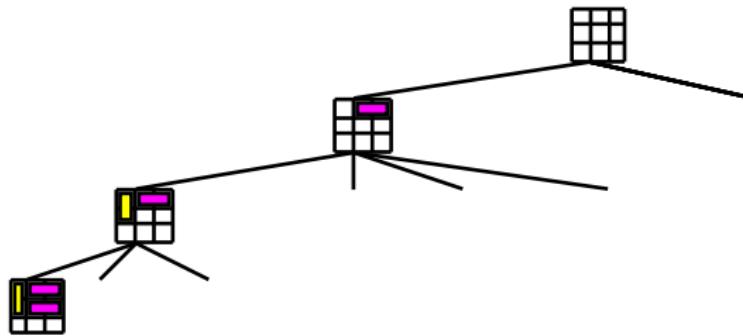
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

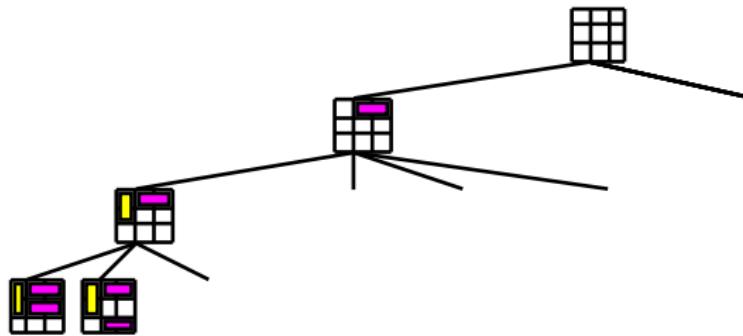
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

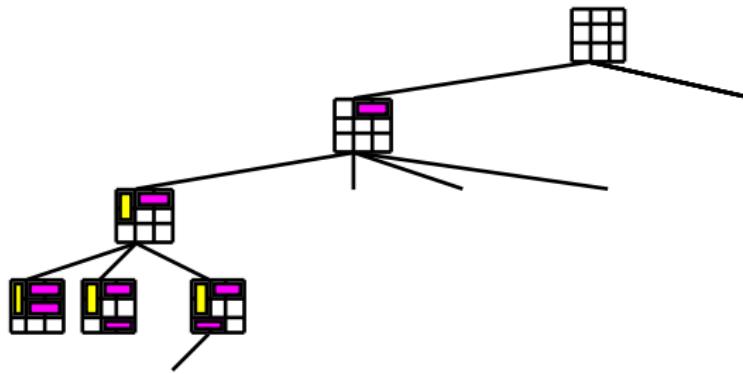
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

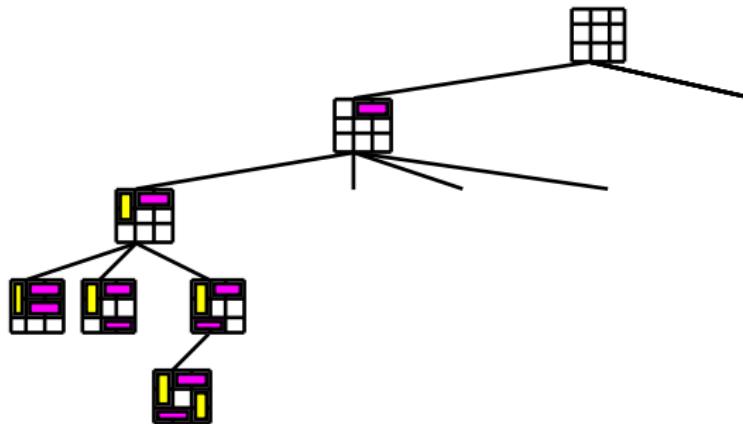
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

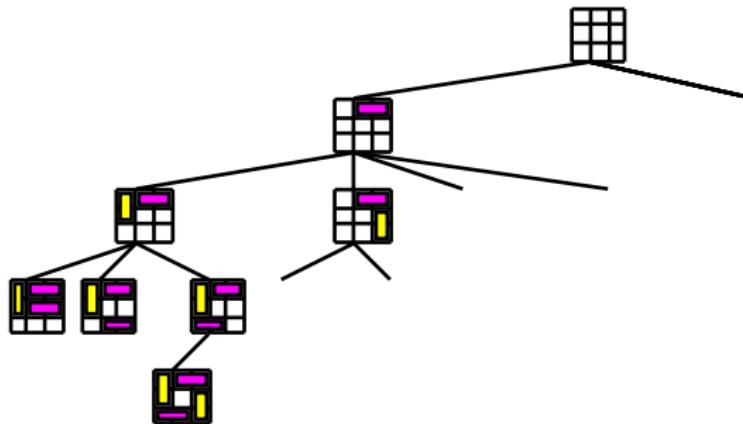
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

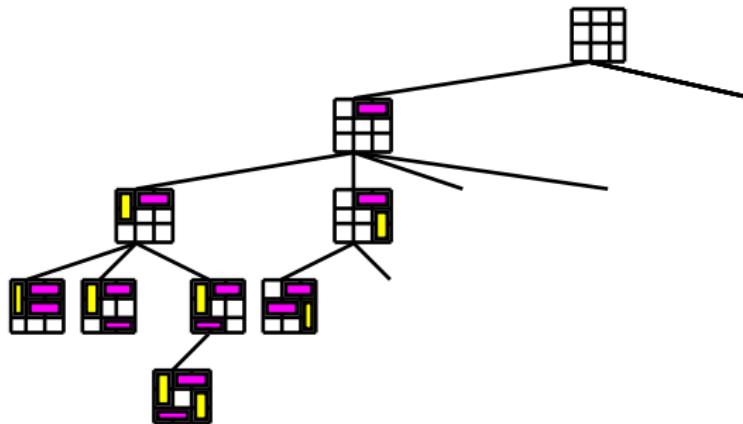
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

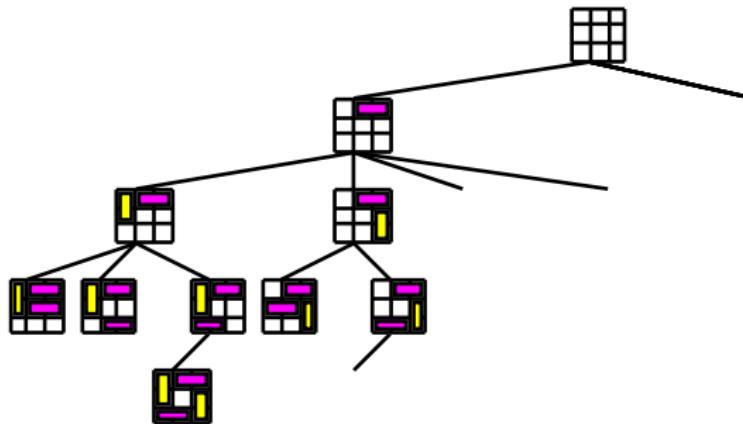
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

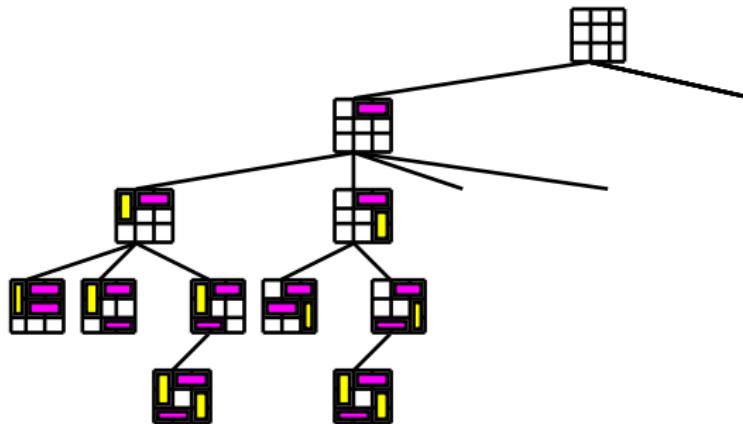
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

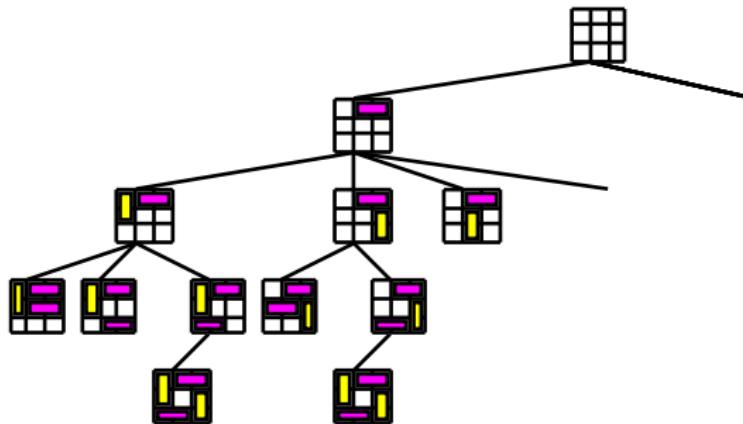
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

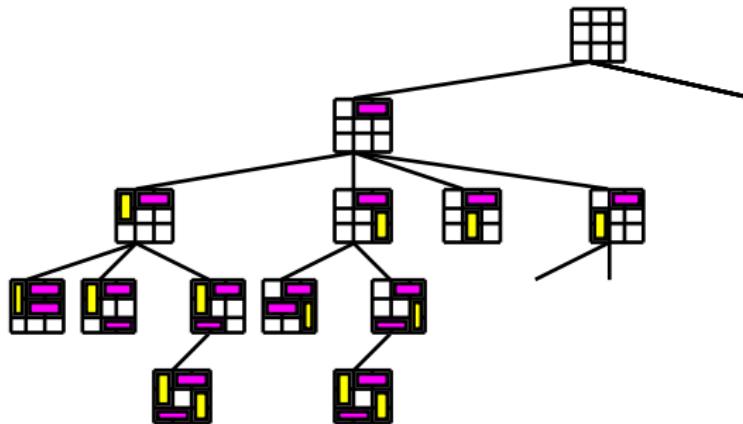
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

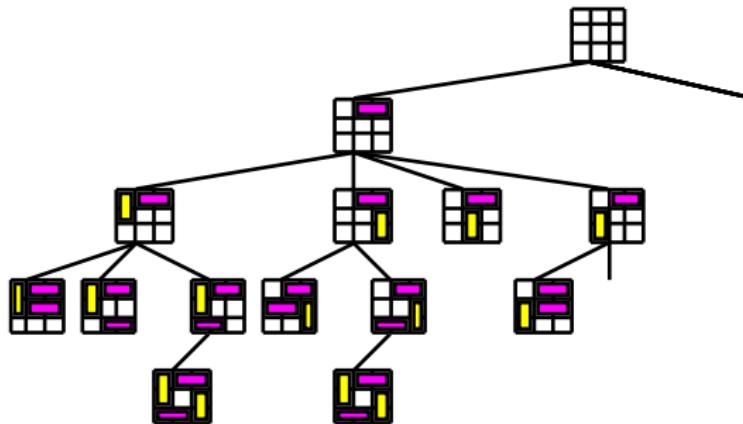
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

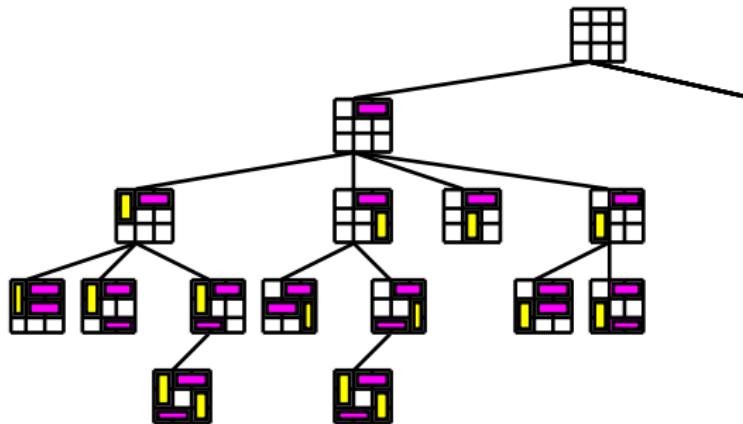
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

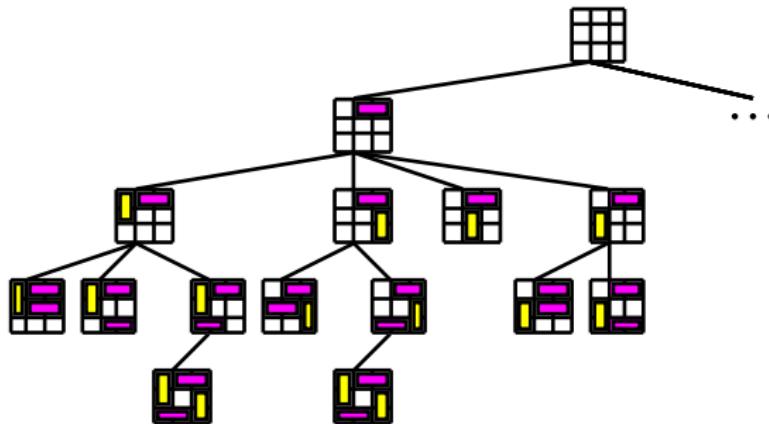
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

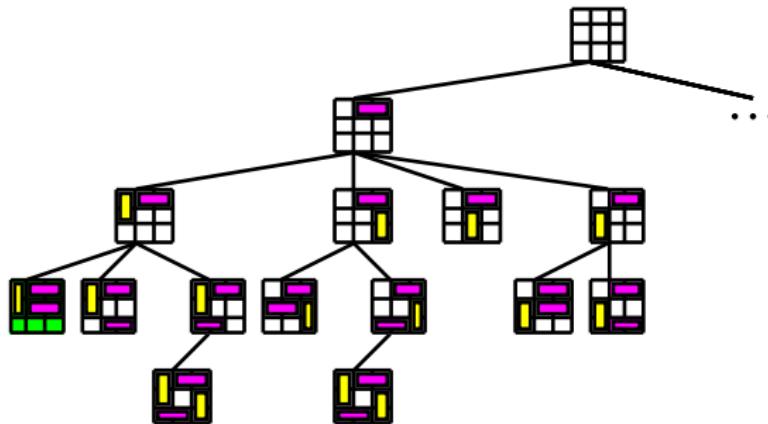
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

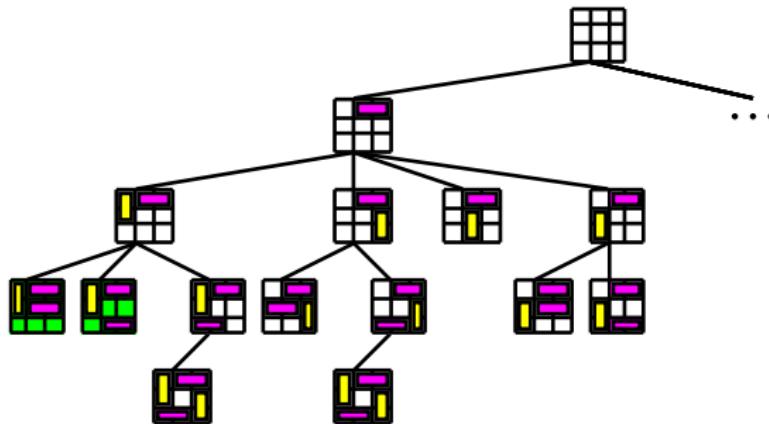
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

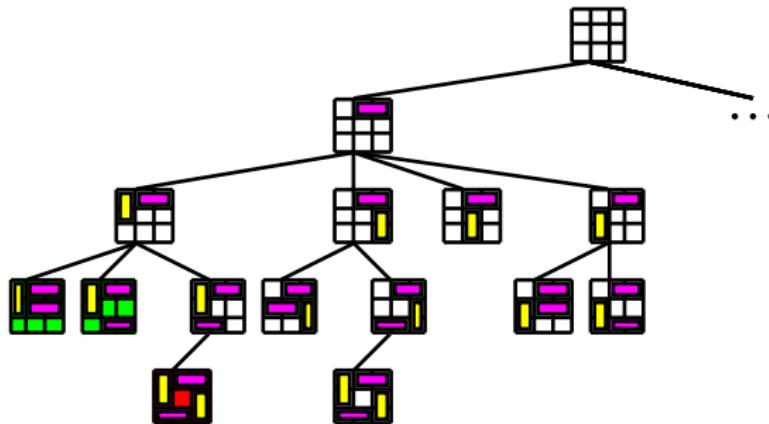
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

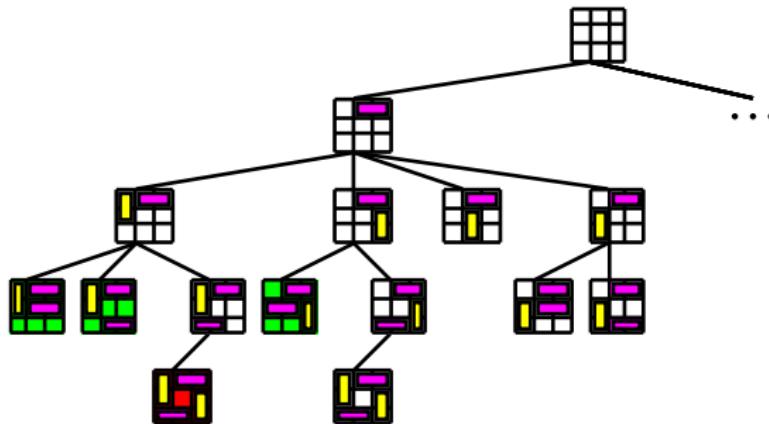
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

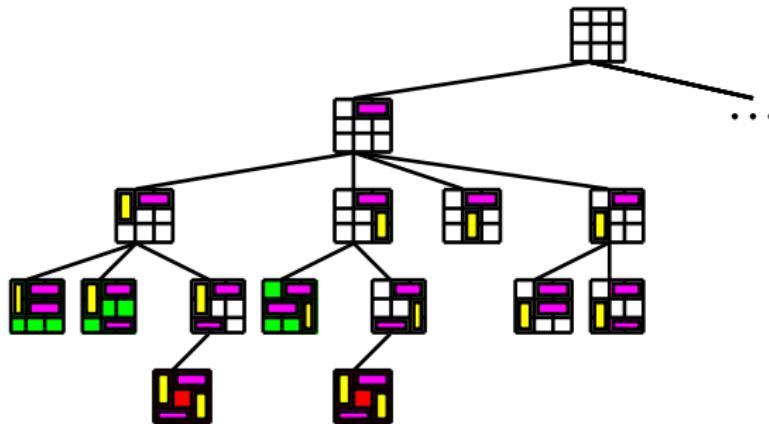
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

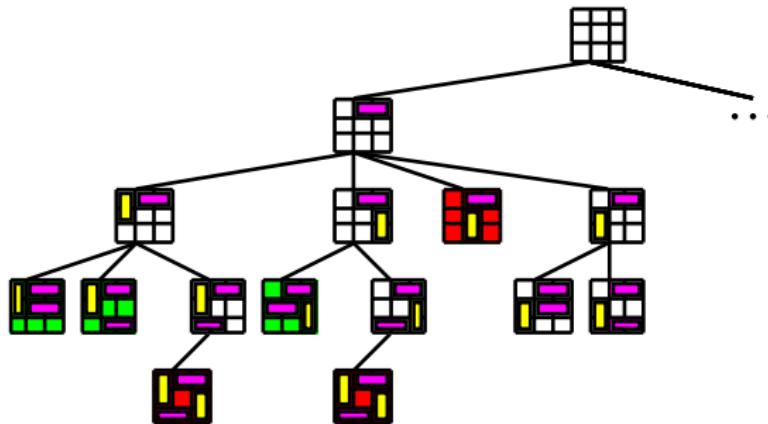
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

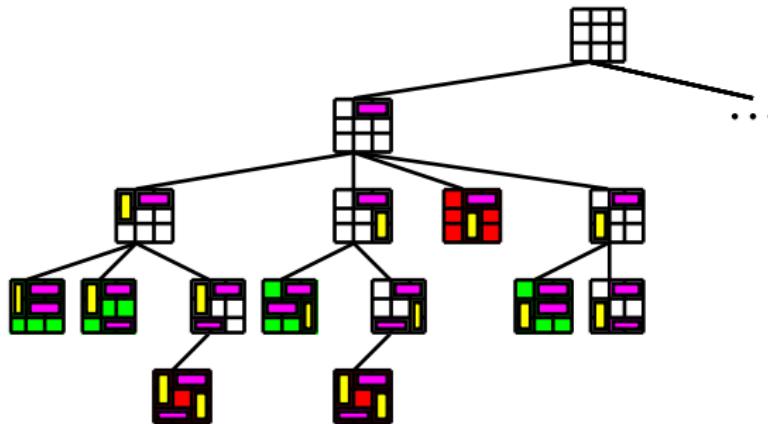
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

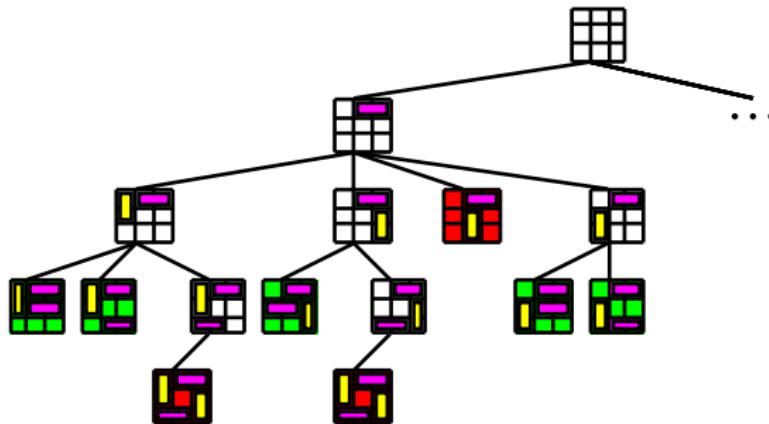
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

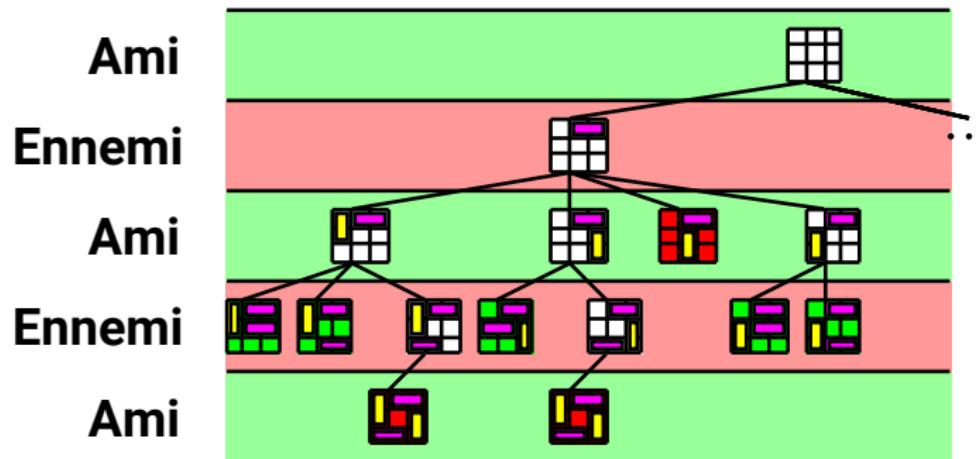
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

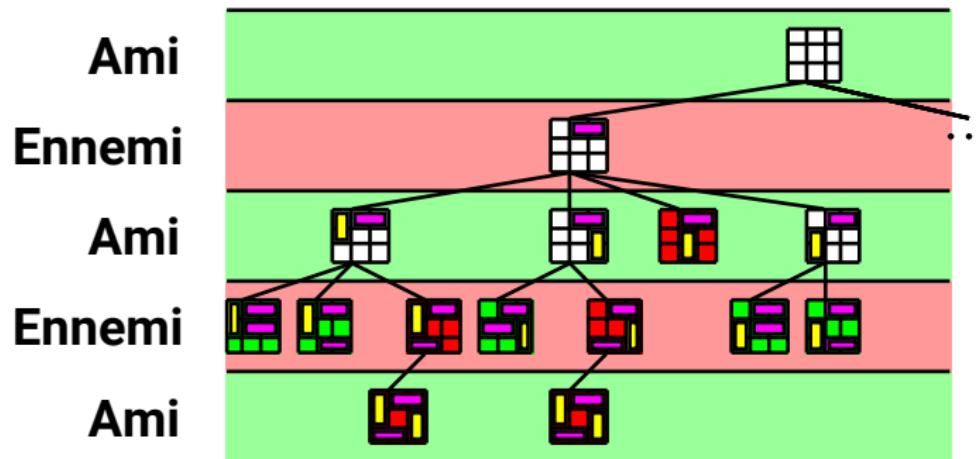
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

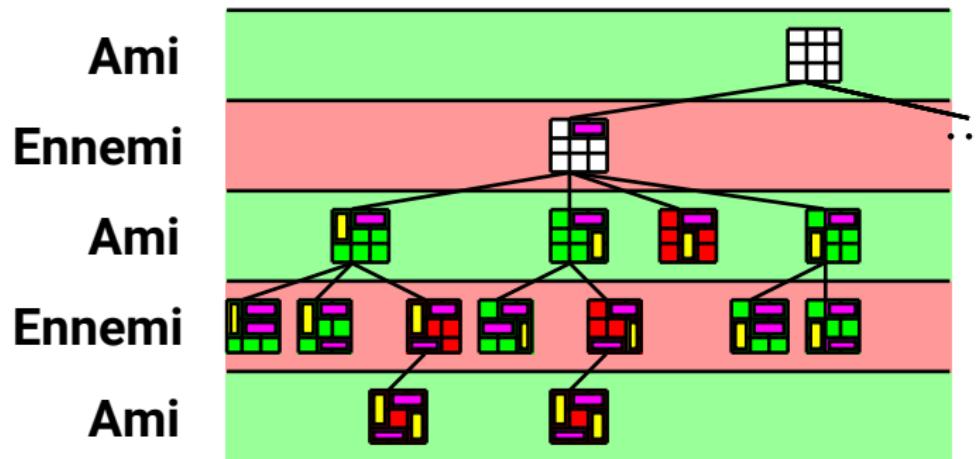
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

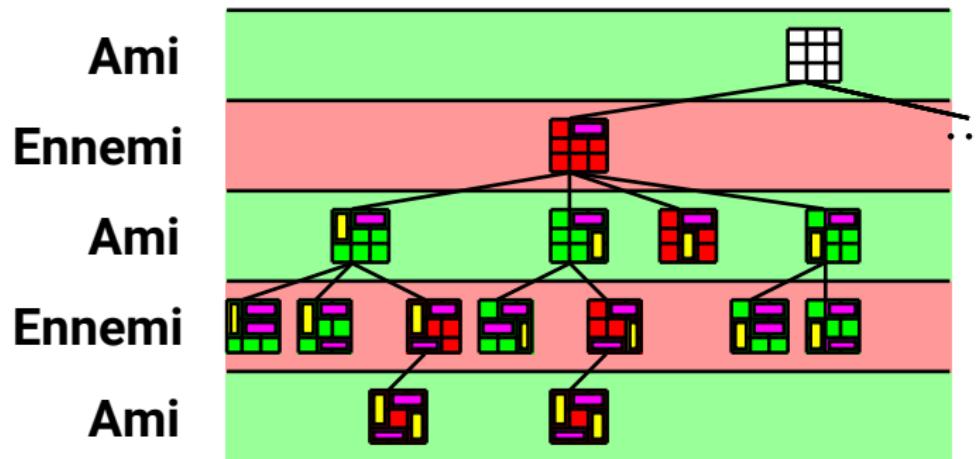
La simulation de tous les coups possibles depuis la position initiale permet de construire *l'arbre de jeu*.



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

Exploration de l'arbre de jeu

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



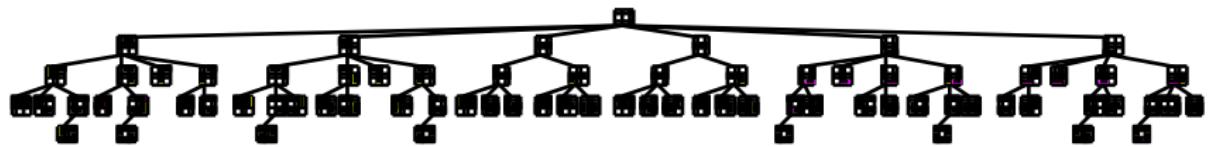
Si Ennemi joue bien, jouer ce coup en premier me fera perdre

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

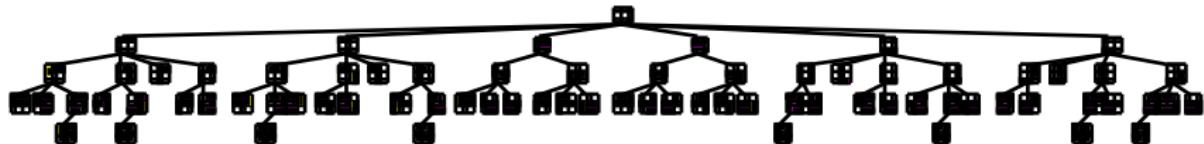
L'exploration peut se voir comme un parcours classique en profondeur à main gauche dans un arbre décrit *en intention*. Dans l'exemple on a trouvé une **Stratégie Gagnante** en 75 noeuds développés.

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

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

Analyse

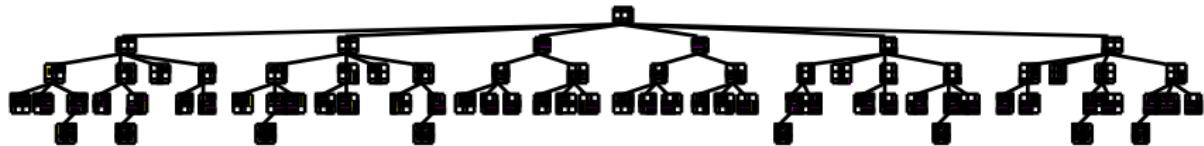
L'exploration peut se voir comme un parcours classique en profondeur à main gauche dans un arbre décrit *en intention*. Dans l'exemple on a trouvé une **Stratégie Gagnante** en 75 noeuds développés.

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	-	-

On se trouve face à une véritable explosion combinatoire

Comment arriver à aller plus loin ?

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 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 ?

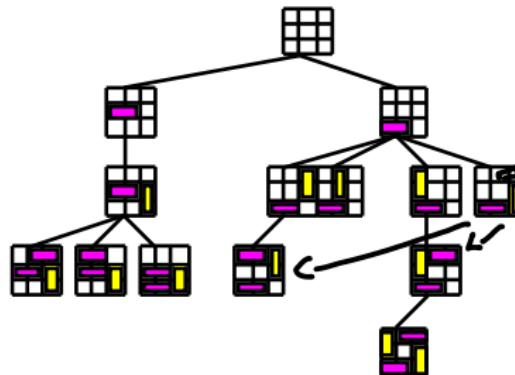
Exploration d'un graphe de jeu

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 !

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 profitant 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 !

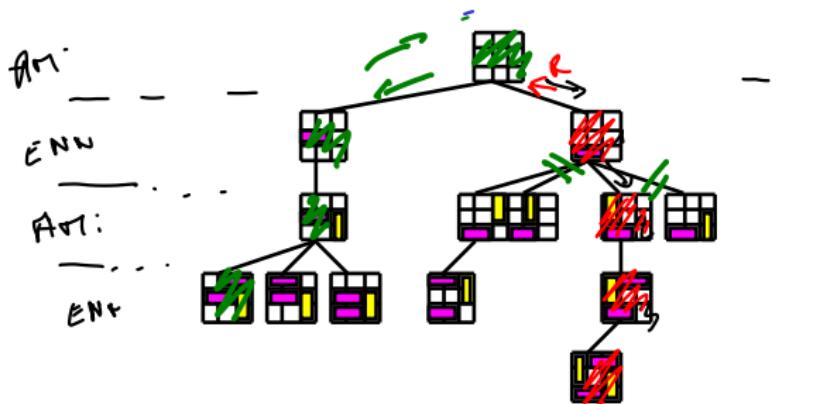
Une première recherche *intelligente*

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 voisines.



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

Introduction
oooooooooooo

Développement Total
oooooooooooooooooooo

Recherche avec Horizon
oooooooooooooooooooo

Bonus
oooo

Introduction
oooooooooooo

Développement Total
oooooooooooooooooooo

Recherche avec Horizon
oooooooooooooooooooo

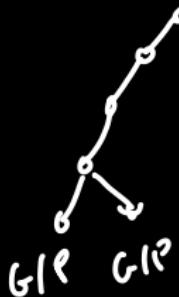
Bonus
oooo

Introduction
oooooooooooo

Développement Total
oooooooooooooooooooo

Recherche avec Horizon
oooooooooooooooooooo

Bonus
oooo



recherche
avec horizon

Introduction d'heuristiques

Idée

On ne développe l'arbre de recherche que jusqu'à une certaine profondeur maximale p . Les feuilles de l'arbre ne sont plus forcément des positions finales du jeu.

On ne peut donc plus se contenter de l'information « gagné » ou « perdu ». Il faut **évaluer** les positions.

Définition intuitive

Les heuristiques sont des **fonctions statiques** associant un **réel** aux **plateaux de jeu**. Dans les jeux avec adversaire, plus l'heuristique est grande et positive, plus on est proche de la victoire. Plus elle est négative et petite, plus on est proche de la défaite.

Exemple des dominos 3×3

Idée

- ➊ On développe l'arbre (ou graphe) de recherche jusqu'à une profondeur donnée (déterminée suivant le temps que l'on a pour jouer et le facteur de branchement estimé du jeu)
- ➋ On évalue la position aux feuilles (dépend du plateau et du joueur)
- ➌ On fait remonter l'évaluation pour trouver le meilleur coup garantissant au moins l'estimation de la feuille

Exemple des dominos 3×3

Idée

- On développe l'arbre (ou graphe) de recherche jusqu'à une profondeur donnée (déterminée suivant le temps que l'on a pour jouer et le facteur de branchement estimé du jeu)
- On évalue la position aux feuilles (dépend du plateau et du joueur)
- On fait remonter l'évaluation pour trouver le meilleur coup garantissant au moins l'estimation de la feuille

Exemple des dominos 3×3

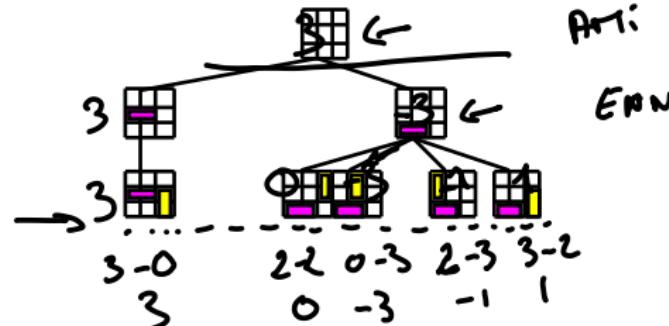
Idée

- On développe l'arbre (ou graphe) de recherche jusqu'à une profondeur donnée (déterminée suivant le temps que l'on a pour jouer et le facteur de branchement estimé du jeu)
- On évalue la position aux feuilles (dépend du plateau et du joueur)
- On fait remonter l'évaluation pour trouver *le meilleur coup garantissant au moins l'estimation de la feuille*

Exemple des dominos 3×3

Idée

- On développe l'arbre (ou graphe) de recherche jusqu'à une profondeur donnée (déterminée suivant le temps que l'on a pour jouer et le facteur de branchement estimé du jeu)
- On évalue la position aux feuilles (dépend du plateau et du joueur)
- On fait remonter l'évaluation pour trouver *le meilleur coup garantissant au moins l'estimation de la feuille*



Exemple d'heuristiques : les Échecs dès 1950

- (1) The relative values of queen, rook, bishop, knight and pawn are about 9, 5, 3, 3, 1, respectively. Thus other things being equal (!) if we add the numbers of pieces for the two sides with these coefficients, the side with the largest total has the better position.
- (2) Rooks should be placed on open files. This is part of a more general principle that the side with the greater mobility, other things equal, has the better game.
- (3) Backward, isolated and doubled pawns are weak.
- (4) An exposed king is a weakness (until the end game).

These and similar principles are only generalizations from empirical evidence of numerous games, and only have a kind of statistical validity. Probably any chess principle can be contradicted by particular counter examples. However, from these principles one can construct a crude evaluation function. The following is an example: -

$$f(P) = 200(K-K') + 9(Q-Q') + 5(R-R') + 3(B-B'+N-N') + (P-P') - \\ [0.5(D-D'+S-S'+I-I') + \\ 0.1(M-M')] + \dots$$

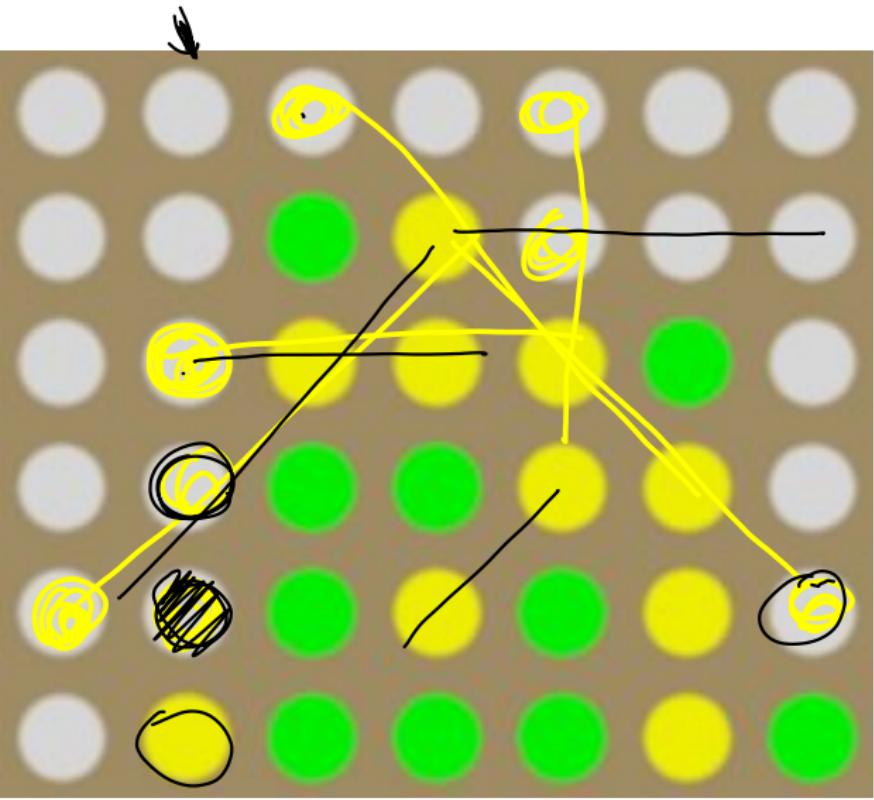
in which: -

- (1) K, Q, R, B, B, P are the number of White kings, queens, rooks, bishops, knights and pawns on the board.
- (2) D, S, I are doubled, backward and isolated White pawns.
- (3) M = White mobility (measured, say, as the number of legal moves available to White).

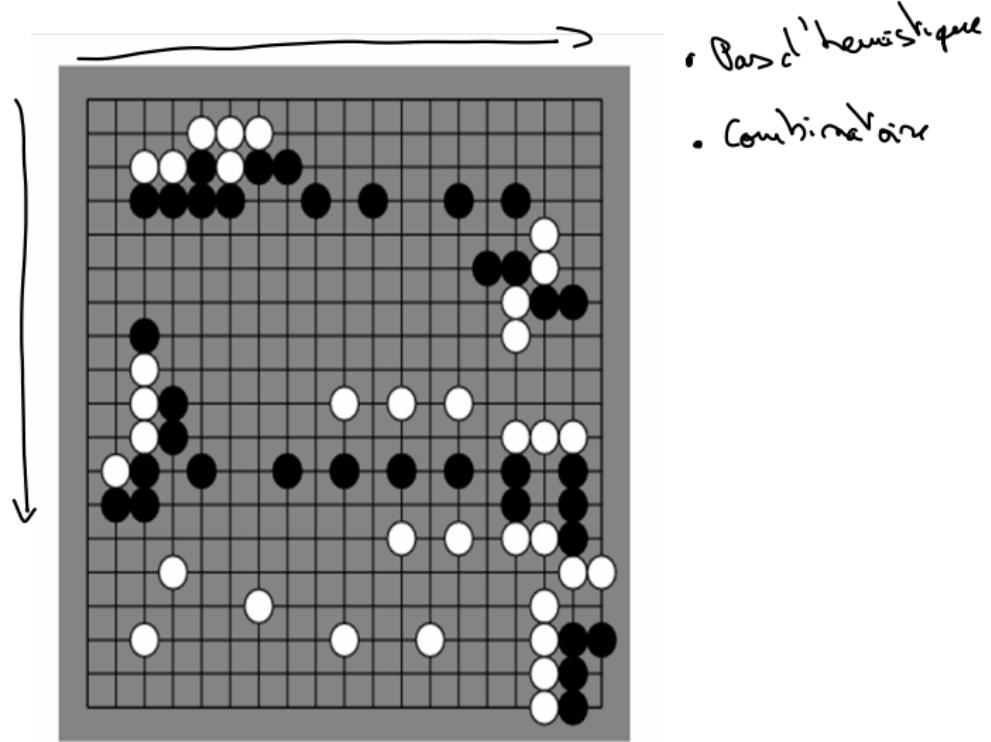
Forces en présence

Exemple d'heuristiques : Puissance 4

Alignement de 4 pions dans la diagonale
 $1 \times (\text{Coef}_1^{(1)})$
 $\otimes 1 \times (\text{Coef}_2^{(10)})$
 $\times (\text{Coef}_3^{(100)})$
+ liberte.



Exemple d'heuristiques : Go



Introduction d'heuristiques

Nouveau but

Trouver la branche permettant de maximiser la valeur heuristique obtenue sur le plateau après les p prochains coups.

Supposition forte

On suppose que les deux joueurs jouent « bien » s'ils suivent les indications de l'heuristique.

En pratique, il faudra prendre en compte cela pour des raisons de performances. À chaque feuille de l'arbre, on devra calculer sa valeur heuristique

Problème : Comment faire remonter la meilleure feuille à la racine ?

Introduction d'heuristiques

Nouveau but

Trouver la branche permettant de maximiser la valeur heuristique obtenue sur le plateau après les p prochains coups.

Supposition forte

On suppose que les deux joueurs jouent « bien » s'ils suivent les indications de l'heuristique.

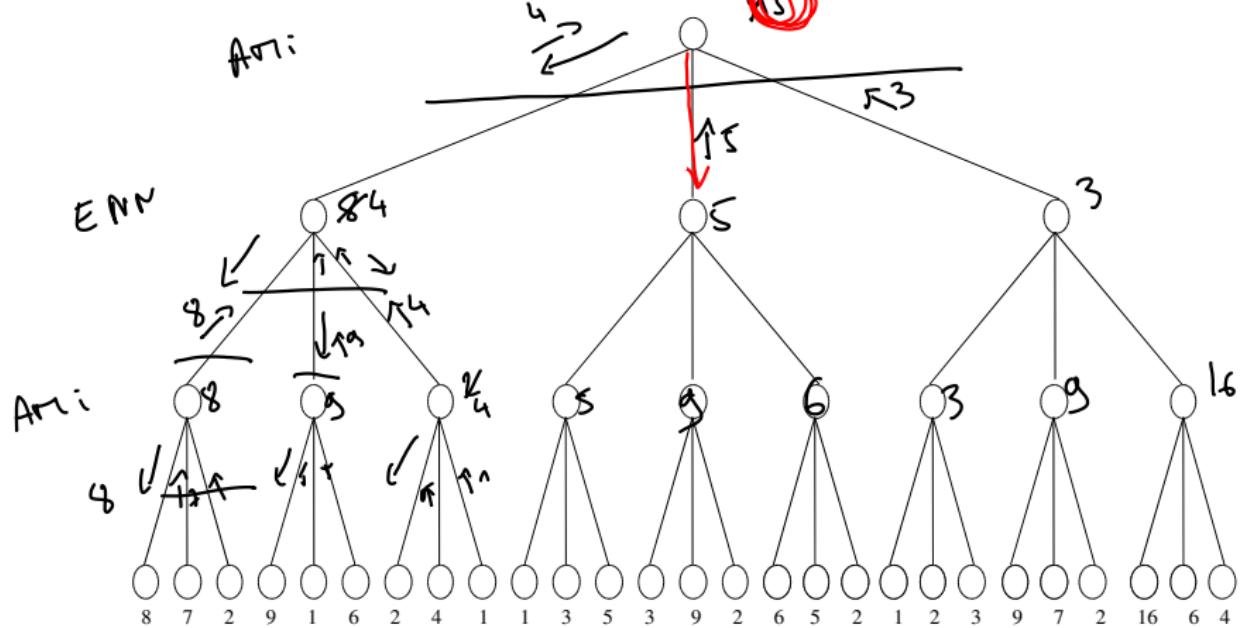
En pratique, il faudra prendre en compte cela pour des raisons de performances. À chaque feuille de l'arbre, on devra calculer sa valeur heuristique

Problème : Comment faire remonter la meilleure feuille à la racine ?

Exemple d'arbre de jeu

△ ut arbres n'existe pas

→ valeur minimax.



Algorithme MiniMax – Fonction *MaxMin*

Idée

On appelle *MaxMin* pour évaluer la valeur MiniMax de la racine. On doit aussi remonter le meilleur coup à jouer qui lui est associé.

```
1: Fonction MaxMin(etat)                                ▷ Évaluation niveau AMI
2:   etat : Plateau de jeu courant

3:   Si EstFeuille(etat) Alors
4:     Retourner evalute(AMI, etat)                      ▷ Évaluation heuristique
5:   Fin Si
6:   Meilleur ← -∞
7:   Pour Tout successeur s de etat Faire
8:     Meilleur ← max(Meilleur, MinMax(s))
9:   Fin Pour
10:  Retourner Meilleur
11: Fin Fonction
```

Algorithme MiniMax – Fonction *MinMax*

Attention

Si l'évaluation de la position a lieu sur un niveau impair (c'est à ENNEMI de jouer), la fonction heuristique doit en tenir compte!

```
1: Fonction MinMax(etat)           ▷ Évaluation niveau ENNEMI
2:   etat : Plateau de jeu courant

3:   Si EstFeuille(etat) Alors
4:     Retourner evalue(ENNEMI, etat)           ▷ Évaluation
heuristique

5:   Fin Si

6:   Pire ← +∞

7:   Pour Tout successeur s de etat Faire
8:     Pire ← min(Pire,MaxMin(s))
9:   Fin Pour

10:  Retourner Pire

Fin Fonction
```

Les deux fonctions MiniMax

Fonction *MaxMin(etat)*

etat : Plateau de jeu courant

▷ Évaluation niveau AMI

Si *EstFeuille(etat)* **Alors**

Retourner *evalue(AMI, etat)*

Fin Si

Meilleur $\leftarrow -\infty$

Pour Tout successeur *s* de *etat* **Faire**

Meilleur $\leftarrow \max(\text{Meilleur}, \text{MinMax}(s))$

Fin Pour

Retourner *Meilleur*

Fin Fonction

▷ Évaluation heuristique

Fonction *MinMax(etat)*

etat : Plateau de jeu courant

▷ Évaluation niveau ENNEMI

Si *EstFeuille(etat)* **Alors**

Retourner *evalue(ENNEMI, etat)*

Fin Si

Pire $\leftarrow +\infty$

Pour Tout successeur *s* de *etat* **Faire**

Pire $\leftarrow \min(\text{Pire}, \text{MaxMin}(s))$

Fin Pour

Retourner *Pire*

Fin Fonction

▷ Évaluation heuristique

Limitations des principes MiniMax

Que se passe-t-il si un joueur

- Ne joue pas « bien »?
- Ne joue pas selon l'estimation heuristique?

Problèmes de collaboration :

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Amplification de valeurs heuristiques

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Limitations des principes MiniMax

- ➊ Que se passe-t-il si un joueur
 - Ne joue pas « bien »?
 - Ne joue pas selon l'estimation heuristique?
- ➋ Problèmes de collaboration :

- ➌ Amplification de valeurs heuristiques

Si deux joueurs utilisent la même fonction heuristique et appliquent une fonction monotone sur les valeurs heuristiques, le résultat sera toujours le même.

Si deux joueurs utilisent deux fonctions heuristiques différentes et appliquent une fonction monotone sur les valeurs heuristiques, le résultat sera toujours le même.

Si deux joueurs utilisent deux fonctions heuristiques différentes et appliquent deux fonctions monotones sur les valeurs heuristiques, le résultat sera toujours le même.

- ➍ Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Limitations des principes MiniMax

- ➊ Que se passe-t-il si un joueur
 - Ne joue pas « bien »?
 - Ne joue pas selon l'estimation heuristique?

- ➋ Problèmes de collaboration :

Dilemme du prisonnier

	Silence	Accusation
Silence	0 0	10 -10
Accusation	-10 10	5 5

- ➌ Amplification de valeurs heuristiques

Le résultat est le même que dans le cas précédent, mais avec des scores amplifiés.

- ➍ Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Limitations des principes MiniMax

- ➊ Que se passe-t-il si un joueur
 - Ne joue pas « bien »?
 - Ne joue pas selon l'estimation heuristique?

- ➋ Problèmes de collaboration :

- ➌ Dilemme du prisonnier

		Silence	Accusation
		0 0	10 1
Silence	0 0		
Accusation	1 10	5 5	

- ➍ Amplification de valeurs heuristiques

- ➎ Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Limitations des principes MiniMax

- ➊ Que se passe-t-il si un joueur
 - Ne joue pas « bien »?
 - Ne joue pas selon l'estimation heuristique?

- ➋ Problèmes de collaboration :

- Dilemme du prisonnier

	Silence	Accusation
Silence	0 0	10 1
Accusation	1 10	5 5

- ➌ Amplification de valeurs heuristiques

Un joueur préfère une valeur dans une zone dangereuse à une zone préférée à une valeur presque identique dans une zone plus « amie »

- ➍ Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Limitations des principes MiniMax

- ➊ Que se passe-t-il si un joueur
 - Ne joue pas « bien »?
 - Ne joue pas selon l'estimation heuristique?

- ➋ Problèmes de collaboration :

- Dilemme du prisonnier

	Silence	Accusation
Silence	0 0	10 1
Accusation	1 10	5 5

- ➌ Amplification de valeurs heuristiques

- ➍ Une valeur faible mais isolée dans une zone dangereuse sera préférée à une valeur presque identique dans une zone plus « amie »
- ➎ En pratique cet effet est contre-balancé par les valeurs heuristiques qui sont théoriquement éloignées aux feuilles.
- ➏ Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Limitations des principes MiniMax

- Que se passe-t-il si un joueur

- Ne joue pas « bien »?
 - Ne joue pas selon l'estimation heuristique?

- Problèmes de collaboration :

- Dilemme du prisonnier

		Silence	Accusation
Silence	0 0	10 1	
Accusation	1 10	5 5	

- Amplification de valeurs heuristiques

- Une valeur faible mais isolée dans une zone dangereuse sera préférée à une valeur presque identique dans une zone plus « amie »

- En pratique cet effet est contre-balancé par les valeurs heuristiques qui sont théoriquement éloignées aux feuilles.

- Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Limitations des principes MiniMax

➤ Que se passe-t-il si un joueur

- Ne joue pas « bien »?
- Ne joue pas selon l'estimation heuristique?

➤ Problèmes de collaboration :

- Dilemme du prisonnier

		Silence	Accusation
Silence	0 0	10 1	
Accusation	1 10	5 5	

➤ Amplification de valeurs heuristiques

- Une valeur faible mais isolée dans une zone dangereuse sera préférée à une valeur presque identique dans une zone plus « amie »
- En pratique cet effet est contre-balance par les valeurs heuristiques qui sont théoriquement éloignées aux feuilles.

➤ Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Limitations des principes MiniMax

- Que se passe-t-il si un joueur

- Ne joue pas « bien »?
 - Ne joue pas selon l'estimation heuristique?

- Problèmes de collaboration :

- Dilemme du prisonnier

		Silence	Accusation
		0 0	10 1
Silence	0 0		
Accusation	1 10	5 5	

- Amplification de valeurs heuristiques

- Une valeur faible mais isolée dans une zone dangereuse sera préférée à une valeur presque identique dans une zone plus « amie »
 - En pratique cet effet est contre-balance par les valeurs heuristiques qui sont théoriquement éloignées aux feuilles.

- Le résultat est le même, après n'importe quelle application d'une fonction monotone sur les valeurs heuristiques.

Recherche dans les arbres de jeux

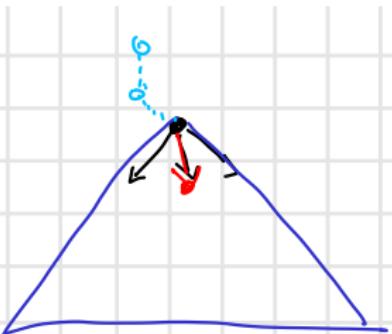
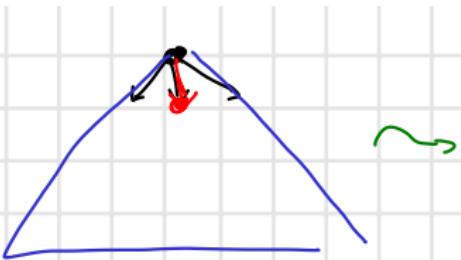
L'espace des états atteignables peut être gigantesque. Il dépend :

- Facteur de branchement b du jeu
- Nombre de coups n d'une partie

Idée des tailles de certains jeux

- Échecs : $p \sim 35$ et $n \sim 30$ coups au minimum
 $|\text{graphe d'états}| = 35^{30}$
Soit 2099139642966190174953146230280399322509765625
- Othello : $5 \leq b \leq 15$
- Go : $b \sim 360$

Il faut retrouver l'idée d'élagage de l'arbre de recherche.



Introduction
oooooooooooo

Développement Total
oooooooooooooooooooo

Recherche avec Horizon
oooooooooooooooooooo

Bonus
oooo

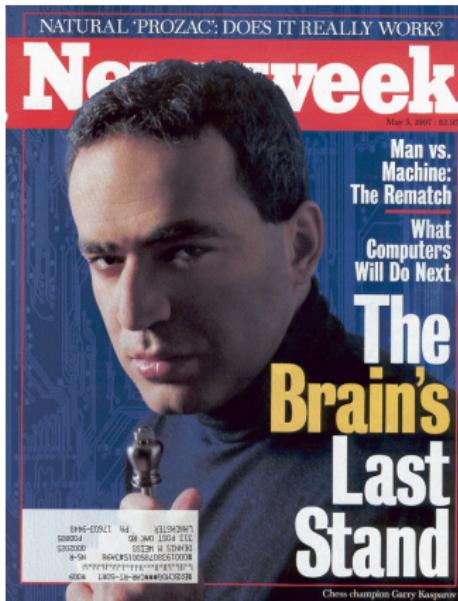
Introduction
oooooooooooo

Développement Total
oooooooooooooooooooo

Recherche avec Horizon
oooooooooooooooooooo●

Bonus
oooo

1997 : Kasparov VS Deep(er) Blue



Kasparov sait que Deep Blue a été grandement amélioré depuis sa défaite de 1996. Il compte « défendre notre dignité » (en tant qu'Humains).

Kasparov vs Deep(er) Blue

Victoire de Deep(er) Blue : 3.5 / 2.5

1er match

Deep Blue perd mais fait un coup "surhumain". Sans gain immédiat.
Vision à long terme du jeu ?

2nd match

Kasparov abandonne brutalement (même si un nul semblait possible)

D'après le Time

(<http://time.com/3705316/deep-blue-kasparov/>) :

"He was again riled by a move the computer made that was so surprising, so un-machine-like, that he was sure the IBM team had cheated"

Kasparov vs Deep(er) Blue

Victoire de Deep(er) Blue : 3.5 / 2.5

1er match

Deep Blue perd mais fait un coup "surhumain". Sans gain immédiat.
Vision à long terme du jeu ?

2nd match

Kasparov abandonne brutalement (même si un nul semblait possible)

D'après le Time

(<http://time.com/3705316/deep-blue-kasparov/>) :

"He was again riled by a move the computer made that was so surprising, so un-machine-like, that he was sure the IBM team had cheated"

Deep(er) Blue : le coup "surhumain"

**Deep(er) Blue a gagné psychologiquement sur son 44ième coup,
1ère manche.**

Comment les ingénieurs d'IBM ont réussi cela ?

- ⌚ Coup à très long terme
- ⌚ Temps de réflexion (anormalement) long

La réponse a été donnée dans le Washington Post

Deep(er) Blue : le coup "surhumain"

C'était un bug