STUCK WIN PROJECT JAVA

# ALGORITHM COMPARISON REPORT

## SAE:101-102

**Berkrouber Benjamin**
**Taskin Semih**
IUTBM-2022
BUT-INFO-S1-C2

# SUMMARY

Rapport de comparaison d'algorithme

# Algorithm n°1 the naive,random

The first algorithm is a naive algorithm which plays with the random notion

## How the algorithm works

**1.**

- The algorithm scans the board and finds the cells which corresponds to the player's color
- Among those squares it saves in a list the squares that are playable
- After that it selects randomly a square in the list which will be the ‘source square’ to play

**2.**

- Then, the algorithm checks the possible moving squares and saves them in a list.
- Finally, it selects randomly in the list a possible moving square which will be the selected square ready to play

## Descriptive scheme

**1.**

Squares that correspond to the player's color

Boxes that are playable

The box selected randomly

**2.**

The source box

Boxes which can be played

The box selected randomly

# Algorithm n°2
# The strategic

The second algorithm is based on a reward strategy according to a box gratification outline

## How the algorithm works

**Strategic AI works by following different steps:**

**1.**

First, it identifies all the possible playable pieces in order to calculate the impact of each pawns. Like that, it determine which one is the best to play.

**2.**

After determining the strongest pawn, the AI will rank the possible moves and calculate the power of each to select the best to play.

**3.**

To allocate a power to a square, the algorithm will sum the scores of the playable squares in front of it under several depth levels.

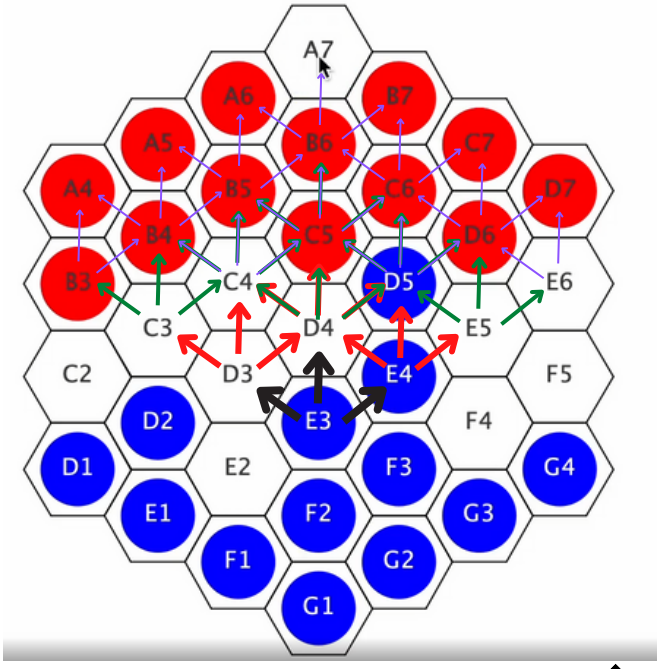
### 1 Operation

The notion of depth means that we will not only observe the 3 possible moves on the next movement.

We are also going to observe the possible moves of each along the three next possibilities.

If we start from one top of the board and go to the bottom we can obtain a maximum depth of 12, so we could look for the moves until the 12th.

This means that 3^12 squares are taken into account for each playable square. This represents a too long calculation time. We will therefore look for a depth of 4 which is a good compromise between power and performance.




### 2 Legend of the diagram

Move to row:

n°1 n°2 n°3 n°4

# 4.

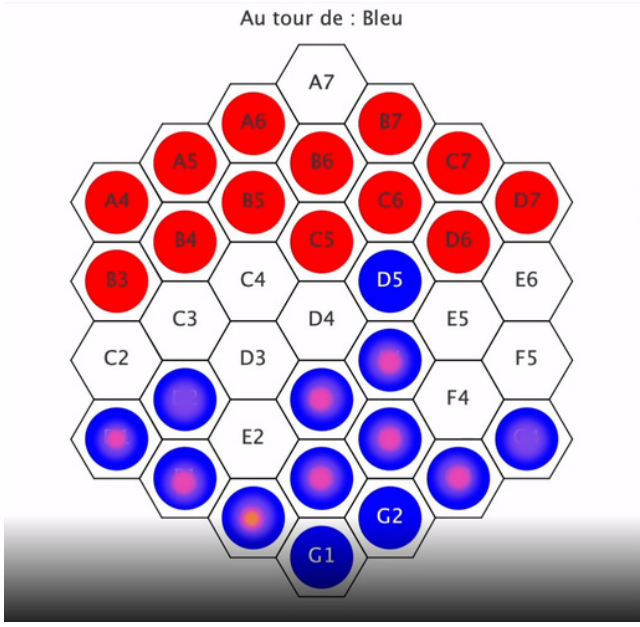
Finally, to determine the score of a square, the AI will take the squares in front or behind into account by adding or subtracting.

## 1 Notion of weighting

**Attribution of the score, power of the box**

To do this, we will determine the value of the box according to its nature.

- Scores ranking
  - The playable square is adjacent to one or more of our colors
  - the playable square is on a border
  - The playable square is adjacent to one or more squares of the opponent's color
  - The playable square is adjacent to one or more free squares, so it is an useless move because it does not block anything.




Playable box  Best box to play

Square blocked by a square of the same color

## 2 Notion of optimization

**Improvement of the score attributed :**

In order to make our AI as efficient as possible we developed adaptations and modulations of the scores attributed according to the different situations.

**Putting depth into perspective.**

The notion of depth means that we move further away with each move in order to observe the game board more and more.

- The further away we go the less impact the square will have, so we divide the score assigned to it by its depth.
- In this way a piece will find it more important to move on squares wearing the same color close to it than to squares wearing the same color further away.
- There is a special case if we look for the index 0. The index 0, will mean that we look for the best square to play and we are going to observe its first 3 squares. It is preferable not to go far from the starting camp. This is why we will put more importance on this move and will not divide it or subtract any value from it. Moreover, it will have a higher coefficient but a common hierarchy than the other depth level.

# Comparison of algorithms

We will now compare the twice algorithms and put forward some conclusions

## Problem: Which algorithm dominates in power & performance.

- We oppose two very different algorithms, one is governed by chance and the other by a strategy.
- We will confront these two AI in order to determine their strength and weaknesses.
- Many questions arise,
  - which one is the most efficient,
  - which one is the most powerful
  - Is there an advantage for the one who starts the game?

## Bonus

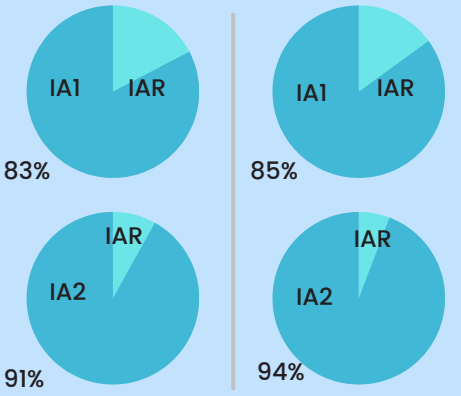
In order to obtain better results in the observation we have used a MIN-MAX AI in order to have an observation on a greater diversity of games. It tries to decrease the number of possible moves to play by increasing the number of moves of the opponent

## Legend

- IA1 = Strategic IA
- IA2 = IA MIN-MAX
- IAR = IA Random

## Affrontements des IA VS IAR

IAR = Joueur B IAR = Joueur R




Observation:

- IA2 wins in more cases than IA1
- The starting player has an advantage that positively affects the game.
- The IAR can never lose 100%.

## Confrontations IA1 VS IA2

Change:

We have 2 AI which have no notion of randomness, so it will play the same game forever. The only change will be the starting player.

- When the strategic AI starts the game, it wins.
- When the MIN-MAX AI starts the game, it loses.

**The AI1 dominates the AI2 no matter what could be the configuration of the game.**

## Comparison of execution resources

We have 2 different algorithms in strategy. However, let's look for a notion of depth in both of them. The more it increases the more the algorithm will take time to execute

| ALGORITHME | 4 | 12 |
|---|---|---|
| IA1 | 1.27101E-4 s | 0.111097649 s |
| IA2 | 1.94501E-4 s | 15.02981132 s |

We notice that it is not playable to use a great depth. and that the AI1 is much more powerful than the AI2

## In conclusion

- The player who start the game is going to have an advantage.
- An IA with a notion of randomness could never lose 100% of the games.
- We can establish a power relationship between the AIs. With IA2 winning in more cases against a random IA, we can determine that it would win in more games against a starting player. However the IA1 beating the IA2 we can only determine that it would win against an advanced player.