

Algorithms comparison report



17/12/2021



IUT
Belfort-
Montbéliard



INFO
Département Informatique
IUT Belfort-Montbéliard

1 - Ours algorithms of « Artificial Intelligence »

For our game project, we realized 3 types of « AI » algorithms :

1. The first algorithm takes and plays the first available box
2. Then, an algorithm who plays randomly
3. Finally, we made an algorithm with a real strategy (more explanation in the second part).

2 - Algorithm comparison

2.1 - First available box algorithm

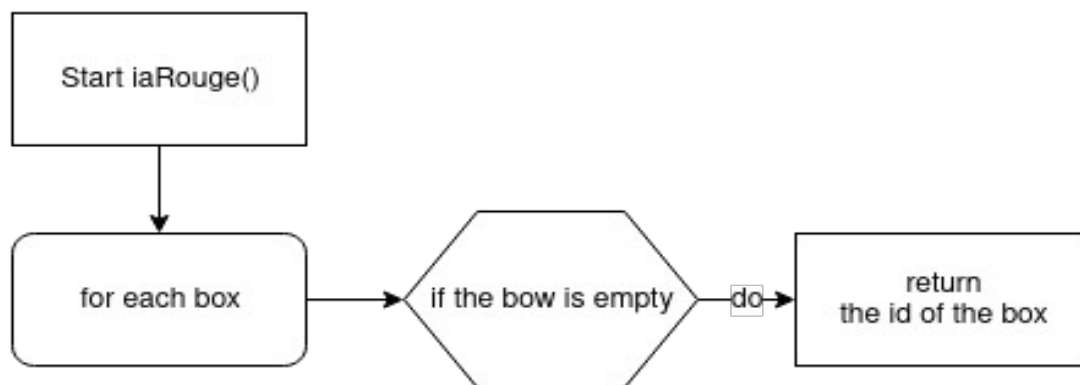


Figure 1: First algorithm diagram

This algorithm cross the boxes of the game, and when it finds an empty box, it returns the box number, else if it doesn't find any empty box, then it returns 0.

2.2 - The “Seeking of the box where the player is loosing the most” algorithm

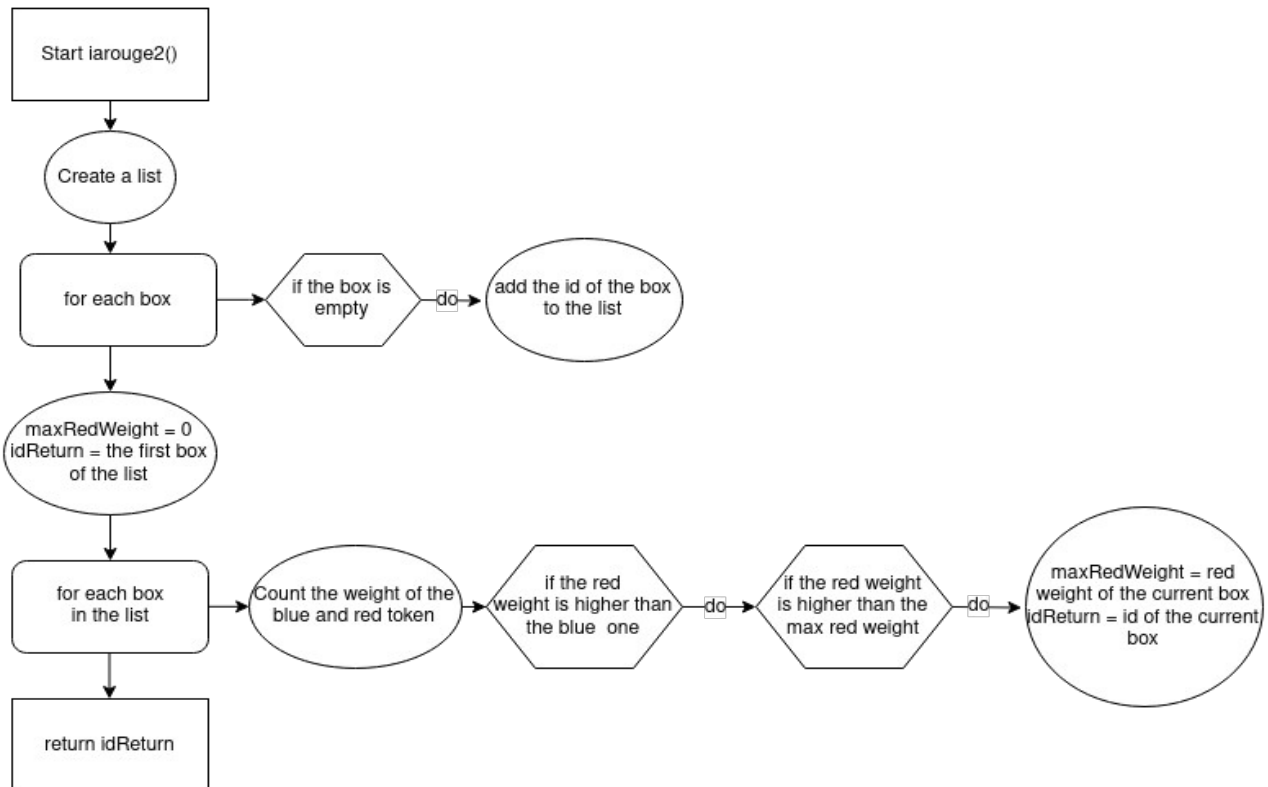


Figure 2: Second algorithm diagram

This algorithm will create a list and seek which boxes are empty and will add the id of empty boxes in the list. Then, for each element in the list : If the sum of the red empty neighboring boxes is higher than the the sum of the blue empty neighboring boxes and higher than the maximum, then the maximum (*maxRedWeight*) equals to the sum of the red empty neighboring boxes, and it keeps the value of the id related to this box.

At the end, it then returns the idReturn (so the most high sum of all boxes) .

2.3 - Comparison between the « First available box » algorithm and the « Seeking of the box where the player is losing the most » algorithm

The first algorithm is very basic : return the first empty box is a very simple strategy, it doesn't has clear objective : he is linear. The second one choose the box where he is losing : it try not to loose.

We have to compare them :

	First algorithm	Second algorithm
Advantages	<ul style="list-style-type: none">• Easy to code• Very fast	<ul style="list-style-type: none">• More complex than the first algorithm• Less predictable• It adapts to the game of the other player
Disadvantages	<ul style="list-style-type: none">• Too predictable• Too simple• It doesn't adapts to the game of the other player	<ul style="list-style-type: none">• Less fast than the first algorithm• More difficult to code

To win the game, the sum of player's neighboring boxes must be lower than the opponent's. Choosing the box where the player has the most chances to win (sum of player's neighboring higher than than the opponent's) is then more efficient, a real strategy to increase the game difficulty and enable this algorithm to have more chances to win the game.

To conclude, the first algorithm is a real fast algorithm and easy to code, but it is too much predictable and it is not a real strategy to win the game. The second one is more complex and less predictable, it has the most chances to win the game because it adapts to the game. However, this algorithm is more difficult to code and less fast than the other one.