Puissance 4

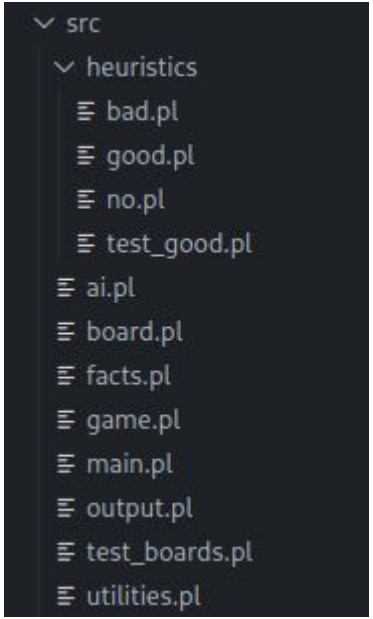
Alix Peigue, Bachir Gueddouda, Ewan Chorynski, Romain Benoit, Théo Choné, Amar Hasan Tawfiq, James Sudlow

Démo

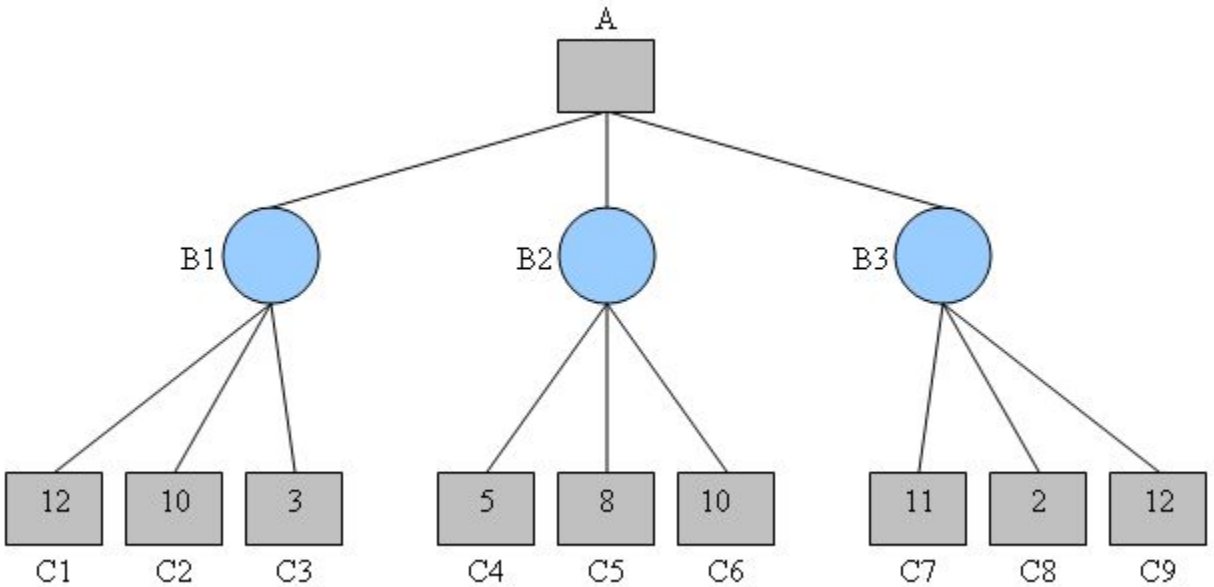
Fonctionnement du programme

Permet des combats JvJ, JvsIA, AIvAI

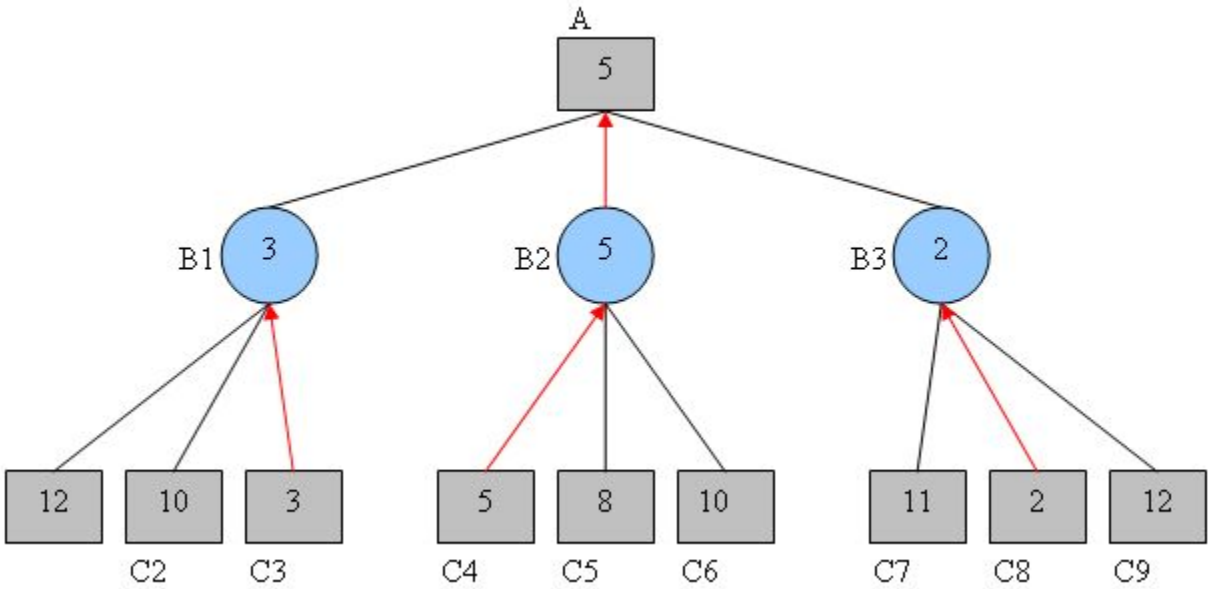
Fonctionnement de l'IA : minmax profondeur 5 avec alpha beta pruning, plusieurs heuristiques possibles + une IA aléatoire



MinMax



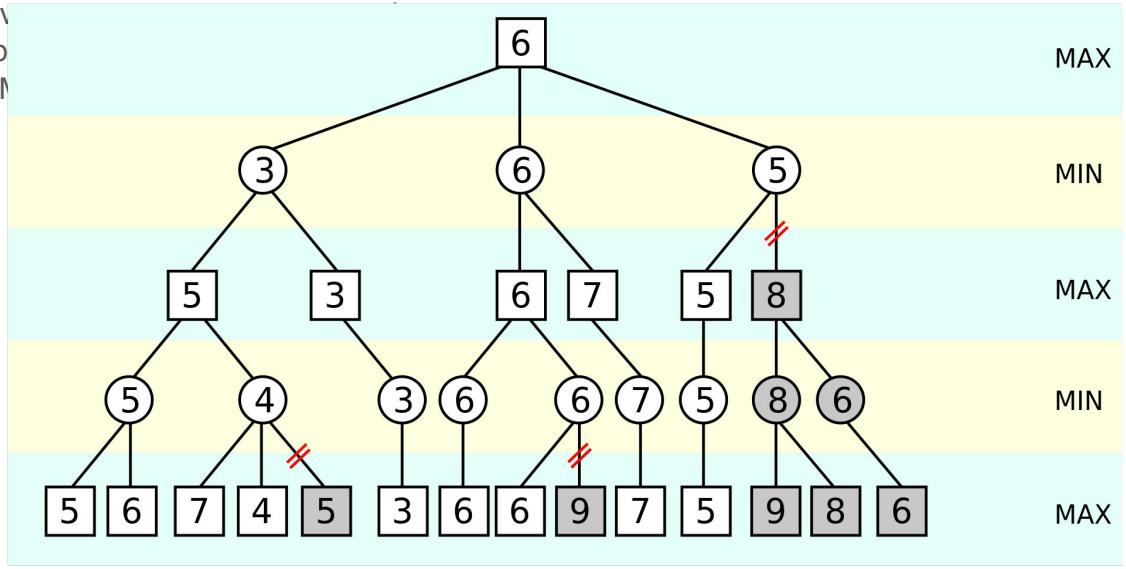
MinMax



Alpha-beta pruning

L'élagage alpha-beta cherche à accélérer le MinMax sans changer son résultat. Pour cela, il réduit le nombre de nœuds dans l'arbre en utilisant le fait qu'un joueur maximise son score et l'autre le minimise. Il y a deux cas :

- lorsqu'on évalue les fils du nœud MIN actuel, on élague si on trouve une valeur de fils inférieure à alpha : en effet, alpha correspond au joueur MAX père du nœud MIN actuel. Sinon, alpha = max(alpha, poids_fils_courant).
- pour les fils du nœud MAX actuel, on élague si on trouv
  de fils supérieure à b
  bêta correspond au joueur M
  MAX
  Sinon, bêta = min(bêta, poids_fils_courant).



Heuristique 1 : solution naïve

La première heuristique est une solution naïve au problème. En puissance 4 le centre a beaucoup de valeur et un quatre en ligne a plus de chance de réussir s'il est plus bas dans le tas. Donc, on calcule la valeur de chaque case donnant:

- une plus grande valeur à ceux au centre
- et ceux vers le bas.

Si la case est occupée par ses pièces, on ajoute la valeur de la case au score, si elle est occupée par une pièce adverse, on enlève la valeur de la case au score.

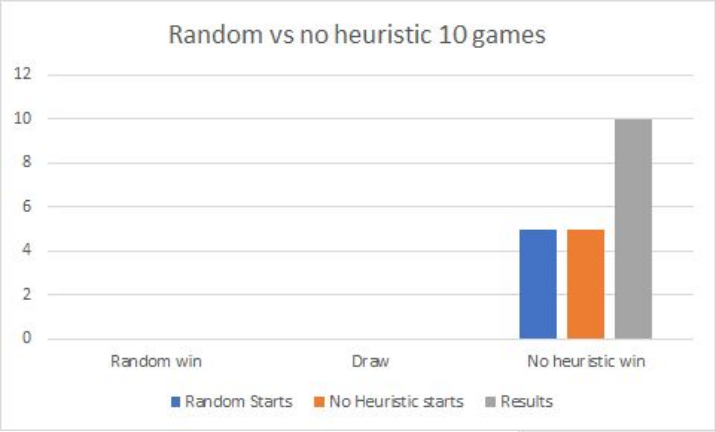
Heuristique 2 : compte des séries

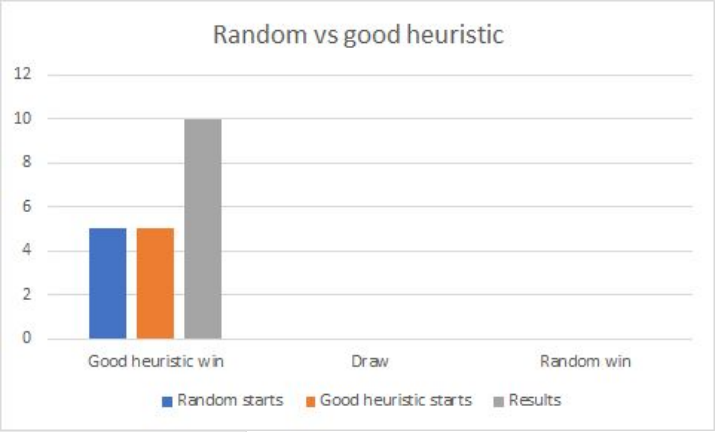
Principe de base : compter les séries, en positif pour nous, en négatif pour les adversaires.

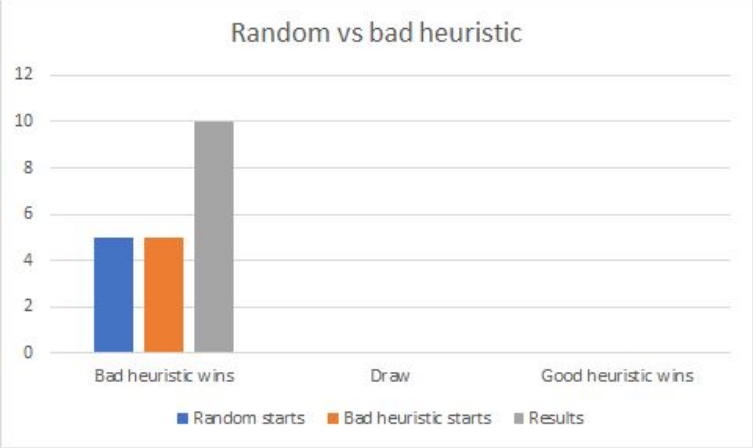
$$\sum_{i \in serie_{joueur}} {l_i}^p - \sum_{i \in serie_{adversaire}} {l_i}^p$$

Dans les exemples suivis, on a p=2

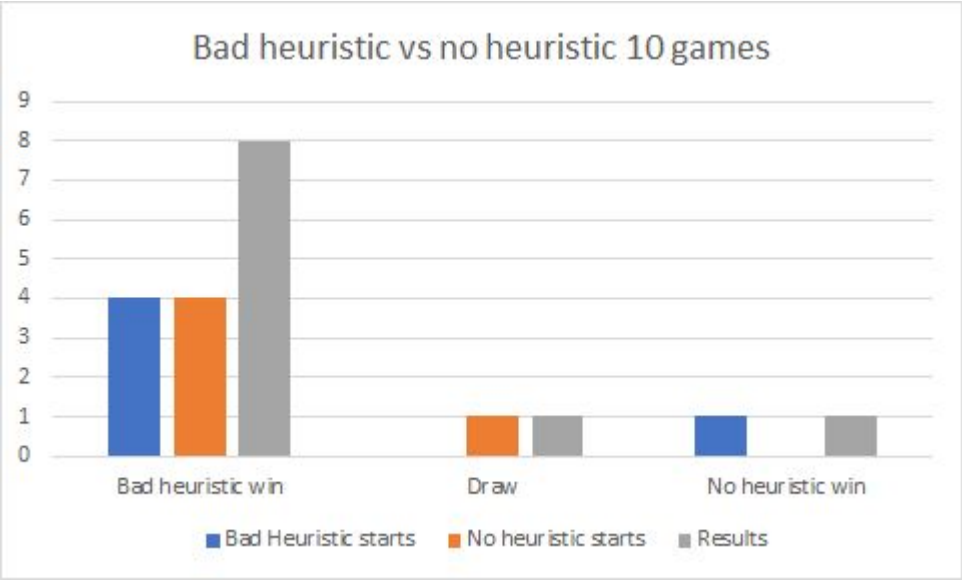
#1 : L’aléatoire perd toujours





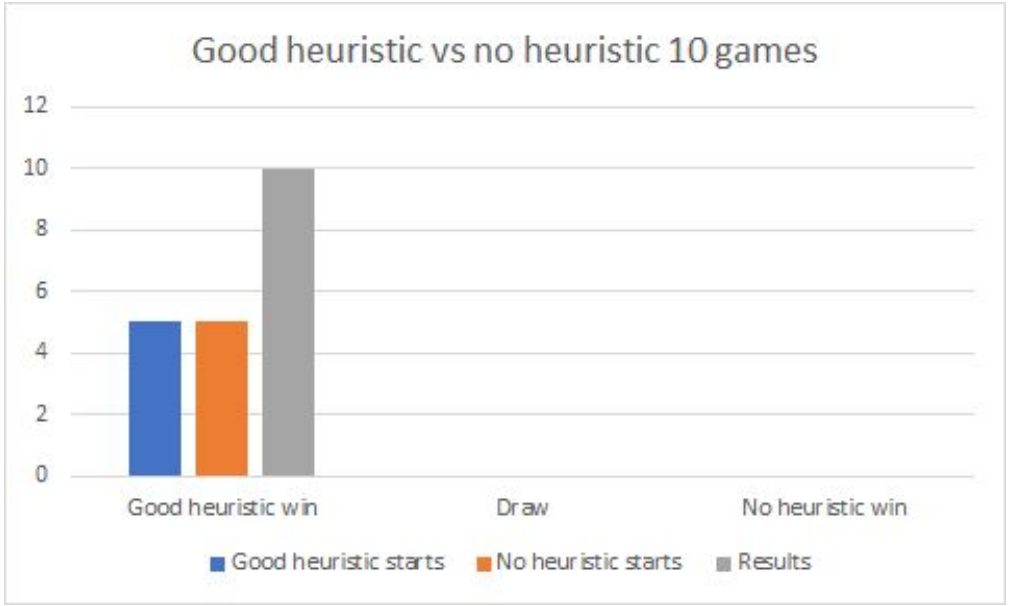


#2 : Bad heuristic vs no heuristic



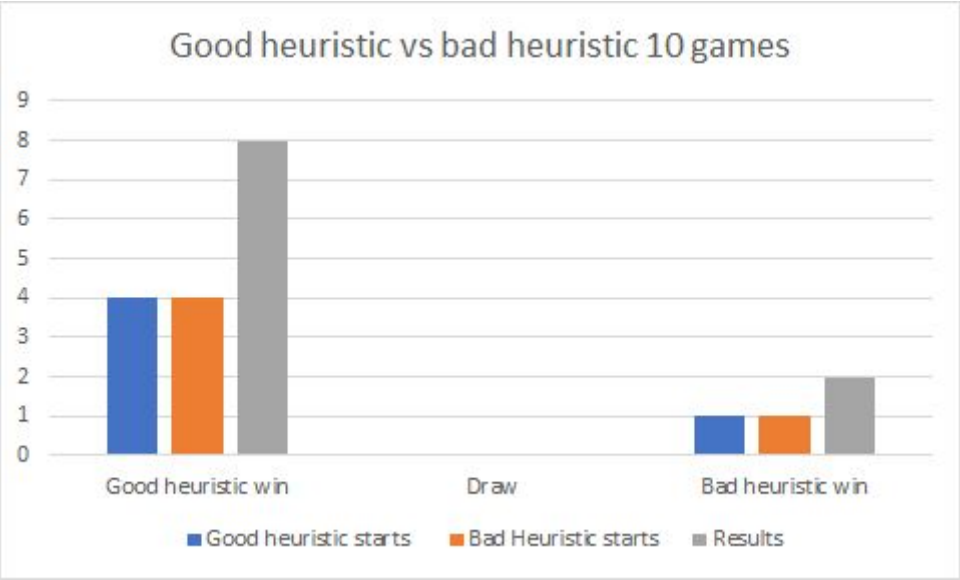
Pour gagner, ils doivent menacer plusieurs lignes gagnantes. Le jeu du bad heuristic centralise et regroupe les pièces et donc augmente la chance de créer du trois en ligne. Tout de même, il y a besoin de chance pour créer une bonne menace.

#3 : Good heuristic vs no heuristic



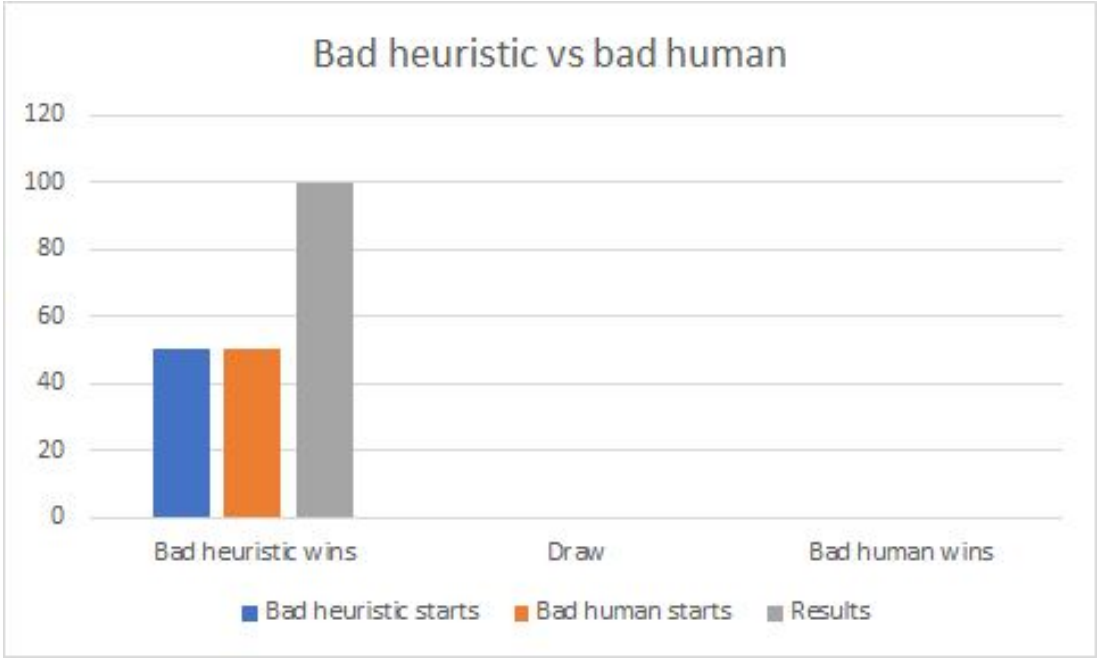
Good heuristic crée des menaces claires et se défend en avance contre les attaques adverses. Le facteur chance disparait.

#4 : Good heuristic vs bad heuristic



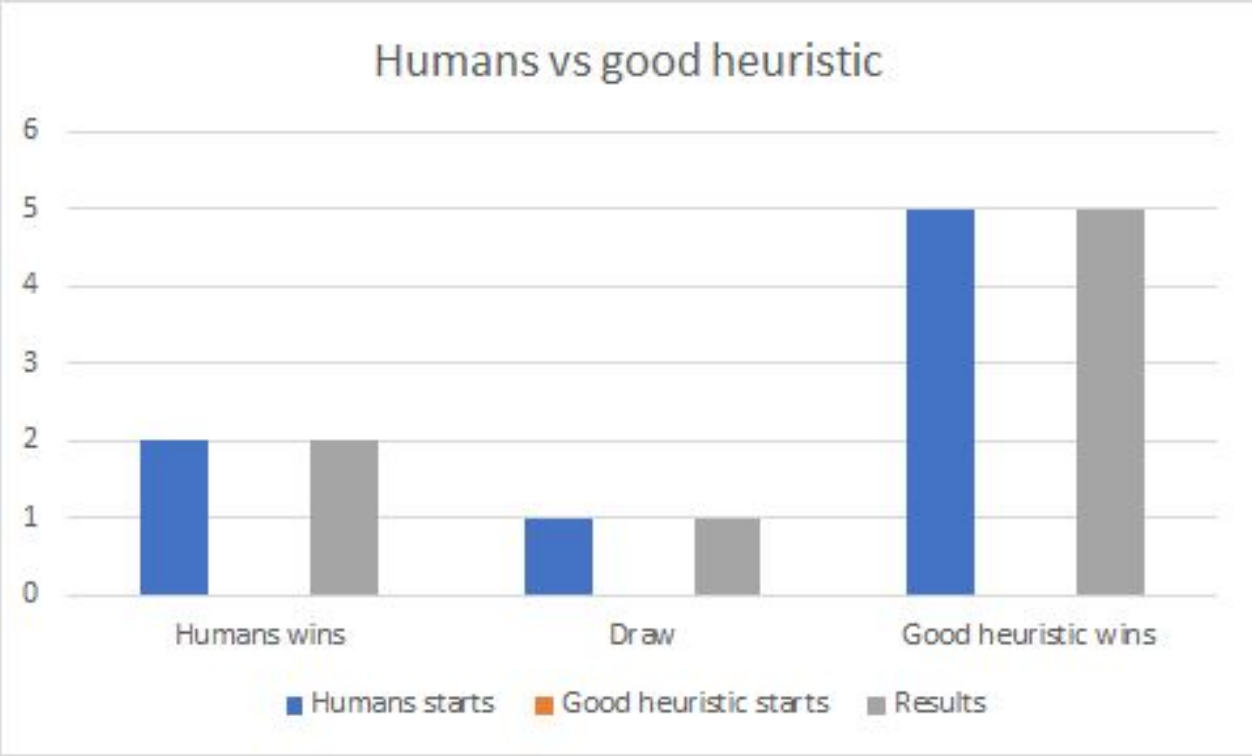
Good heuristic s’adapte au style adverse et gagne souvent sur les bords. Mais avec assez de chance, bad heuristic pouvait avoir trop de menaces au centre.

#5 : Bad heuristic vs bad human (Théo)

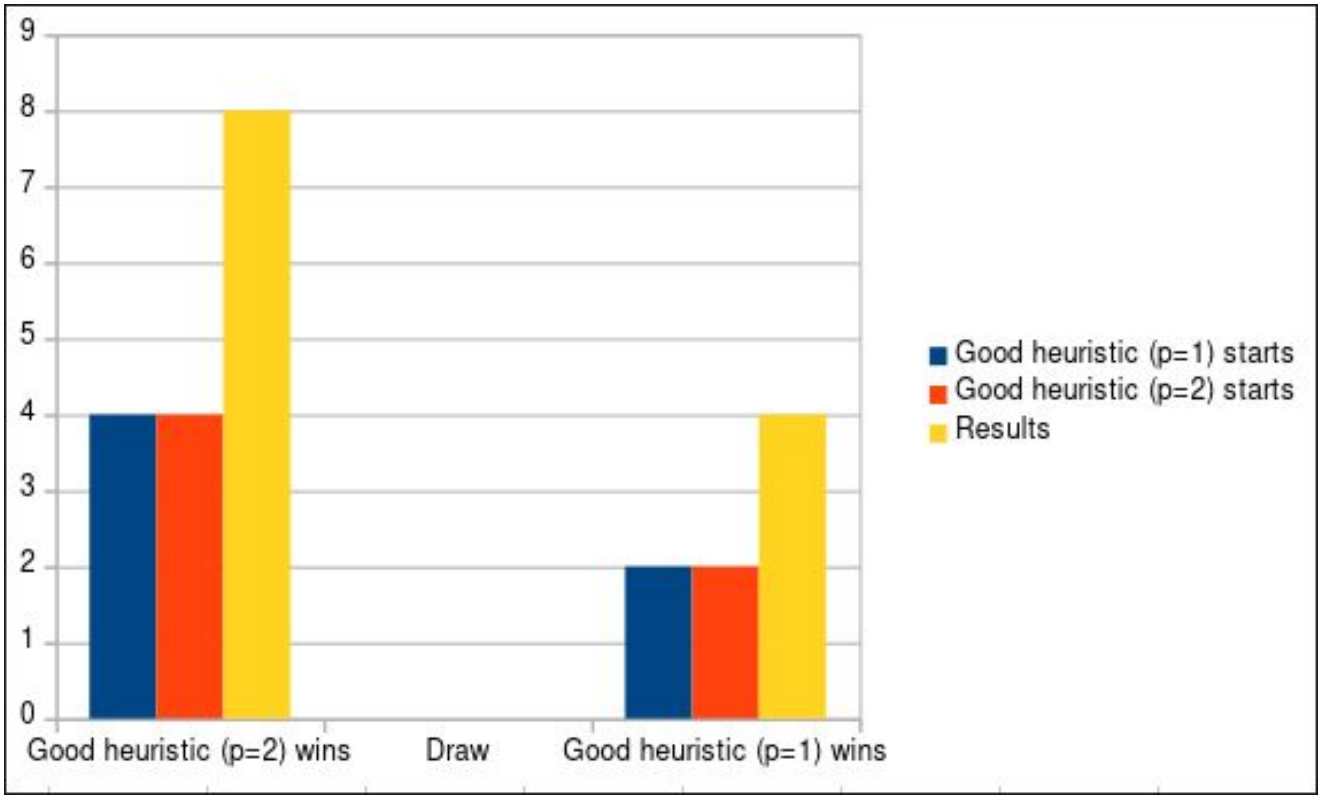


Théo a besoin de s’entraîner un peu

#6 : Good heuristic vs 3 good humans



#7 : Influence de p dans l'heuristique 2



Difficultés rencontrées

Changement de paradigme qui nous empêche d’utiliser les stratégies qui sont presque des automatismes.

Débogage difficile (les traces ne remplacent pas les breakpoints).

Les conventions de nommage du projet d’exemple sont discutables, nous l’avons utilisé et le code est maintenant difficile à maintenir.