

**Compte rendu de TP**

**Intelligence Artificielle**

15 mars 2021

HOK Jean-Rémy

INSA\* Promotion 55, 4ème Année IR\*

135, Avenue de Rangueil

31077 Toulouse Cedex 4

\*INSA : Institut National des Sciences Appliquées

\*IR : Informatique et Réseaux

**Compte rendu de TP**

**Intelligence Artificielle**

15 mars 2021

HOK Jean-Rémy

INSA\* Promotion 55, 4ème Année IR\*

**INSA Toulouse**

135, Avenue de Rangueil

31077 Toulouse Cedex 4

\*INSA : Institut National des Sciences Appliquées

\*IR : Informatique et Réseaux

Table des matières

[Introduction 1](#_Toc66725019)

[TP1 – Algorithme A\* – Application au Taquin 2](#_Toc66725020)

[Familiarisation avec le problème du Taquin 3x3 2](#_Toc66725021)

[Développement des 2 heuristiques 3](#_Toc66725022)

[TP2 – Algo minmax – Application au TicTacToe 5](#_Toc66725023)

# Introduction

L’objectif des deux TPs a été de mettre en pratique nos connaissances sur les algorithmes de résolution de problèmes basés sur la recherche arborescente avec heuristique, à savoir :

* TP1 : Algorithme A\* appliqué au Taquin
* TP2 : Algorithme MinMax / Alpha-Beta appliqué au TicTacToe

# TP1 – Algorithme A\* – Application au Taquin

## Familiarisation avec le problème du Taquin 3x3

#### Quelle clause Prolog permettrait de représenter la situation finale du Taquin 4x4 ?

* final\_state([[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,vide]]).

#### A quelles questions permettent de répondre les requêtes suivantes :

* initial\_state(Ini), nth1(L,Ini,Ligne), nth1(C,Ligne, d).

On récupère la ligne L et la colonne C de la pièce « d » dans l’état initial : à savoir ligne 2 colonne 3.

* final\_state(Fin), nth1(3,Fin,Ligne), nth1(2,Ligne,P).

On récupère la pièce P en position (3,2), ligne 3 colonne 2, dans l’état final : à savoir « f »

#### Quelle requête Prolog permettrait de savoir si une pièce donnée P (ex : a) est bien placée dans U0 (par rapport à F) ?

U0 : état initial et F : état final

initial\_state(Init), nth1(L,Init,Ligne), nth1(C,Ligne, a), final\_state(Fin), nth1(L,Fin,Ligne2), nth1(C,Ligne2,a).

Qui retourne *false* dans notre cas.

#### Quelle requête permet de trouver une situation suivante de l'état initial du Taquin 3×3 (3 sont possibles) ?

initial\_state(Init), rule(\_, \_, Init, Suiv).

#### Quelle requête permet d'avoir ces 3 réponses regroupées dans une liste ?

initial\_state(Init), findall(Suiv, rule(\_,\_,Init,Suiv),Suiv\_Liste).

#### Quelle requête permet d'avoir la liste de tous les couples [A, S] tels que S est la situation qui résulte de l'action A en U0 ?

initial\_state(Init),

findall([Action,Suiv], rule(Action,\_,Init,Suiv),Suiv\_Liste).

### Développement des 2 heuristiques

#### Heuristique 1 : nombre de pièces mal placées

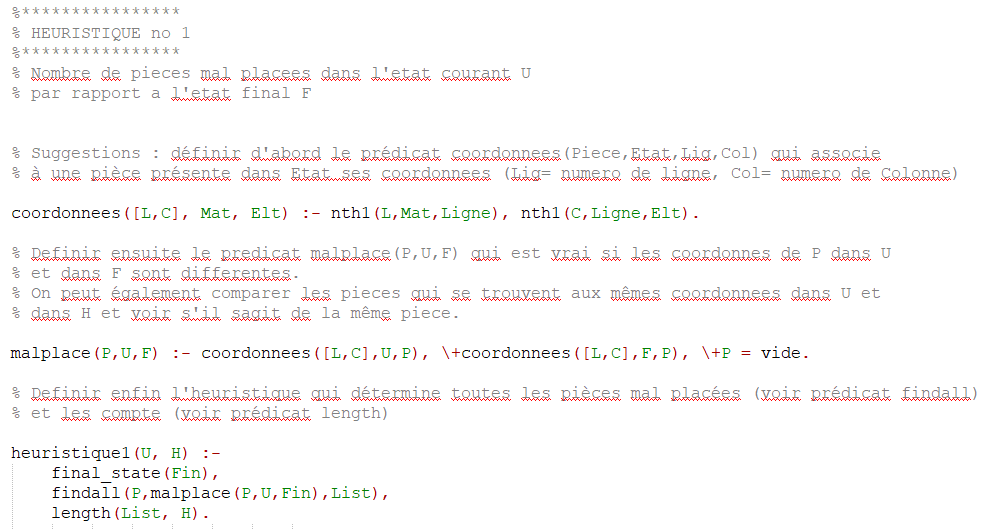


Figure 1 : Heuristique 1

|  |  |
| --- | --- |
| Figure 2 : Enoncé heuristique 1 | Nous trouvons bien 4 lorsque nous appliquons l’heuristique 1 sur l’état initial. |

#### Heuristique 2 : distance de Manhattan

Blabla

Blabla Note de bas de page[[1]](#footnote-1) blabla

# TP2 – Algo minmax – Application au TicTacToe



1. Note de bas de page : Description [↑](#footnote-ref-1)