Intelligence Artificielle – Projet ESCAMPE

# Développement de l’heuristique

L’approche du développement de l’heuristique sera d’améliorer itérativement les performances du joueur automatique contre le joueur aléatoire en ajoutant ou modifiant des caractéristiques de son heuristique.

Les performances seront mesurées par pourcentage de victoires contre le joueur aléatoire. La performance médiocre sera donc celle du joueur aléatoire, soit un pourcentage de victoire de 50%.

## Première idée : Distances licornes-paladins

### Définition

Heuristique Alpha :

Une heuristique simple à implémenter peut être la différence entre la plus courte distance entre un paladin ennemi et la licorne alliée, et la plus courte distance entre un paladin allié et la licorne ennemie. Si cette différence est positive, la licorne ennemie sera plus « attaquée » que la licorne alliée. Cette heuristique ne prend pas en compte les déplacements par liserés : Bien qu’un paladin puisse être plus proche d’une licorne qu’un autre, si les liserés ne lui permettent pas de prendre la licorne, il n’aura pas tellement un avantage conséquent. Mais on reconnaîtra le caractère simple et relativement cohérent de cette heuristique.

### Performances

Un algorithme AlphaBeta à profondeur 5 avec l’heuristique Alpha a un taux de victoires de 60% contre un joueur aléatoire. On en conclut que l’heuristique représente peu correctement l’état d’un plateau. Les heuristiques suivantes devront inclure les types de liserés sur lesquels se trouvent les paladins. L’heuristique ne donne quasiment pas d’information utile sur le placement initial.

### Idées d’amélioration

Prendre en compte les liserés.

## Potentiels attaquants

### Définition

Heuristique Beta :

Pour prendre en compte les liserés, on peut définir une heuristique donnant la différence entre le nombre de paladins alliés pouvant prendre la licorne ennemie (si un paladin est sur un liseré donnant x déplacements et se trouve à une distance x de la licorne ennemie) moins le nombre de paladins ennemis pouvant prendre la licorne alliée.

### Performances

Un algorithme AlphaBeta à profondeur 5 avec l’heuristique Beta a un taux de victoires de 59% contre un joueur aléatoire. On en conclut que l’heuristique représente peu correctement l’état d’un plateau. Si l’heuristique permet de juger assez fiablement un plateau de fin de partie, il ne donne que peu voire aucune indication sur le placement de départ ou sur un plateau de début/milieu de partie.

### Idées d’amélioration

Il serait peu utile d’essayer de fusionner les heuristiques Alpha et Beta, puisque Beta comprend davantage la notion de danger pour une licorne (la licorne n’est pas nécessairement plus mise en danger par un paladin à 1 case qu’à 3 cases si celui-ci peut effectivement la prendre).

## Potentiels attaquants + distance aux paladins

### Définition

Heuristique Gamma :

On définit Gamma comme une heuristique Beta auquel on rajoute les distances entre les licornes et tous les autres paladins adverses. Tels que :

Un bon plateau correspond maintenant en plus à un plateau où la licorne n’est pas menacée directement et où les paladins ennemis sont loins.

### Performances