

# Rapport SAÉ: IA Python

## A. Stratégie de notre IA

**Pour la stratégie du pacman**, nous avons décidé de calculer chaque possibilité où le pacman peut se déplacer, et choisir la meilleure case en fonction des points qu'il obtient, le pacman cherche donc les points et prend le chemin le plus court, si un fantôme est en vue, même si un bonus est à côté à part le glouton, le pacman décidera en priorité de fuir le fantôme. Il fait donc en fonction du fantôme c'est-à-dire qu'en fonction de la distance entre son objectif et le fantôme il fera le choix de soit y aller ou fuir.

**Pour ce qui est de la stratégie du fantôme**, le fantôme chasse le pacman, à part si un glouton est pris, si le pacman à un glouton, le fantôme prend la décision de fuir avec les chemins les plus courts possibles. Lors de la chasse, le fantôme cherche la distance la plus courte pour aller au pacman et si à une distance égale il y a un chemin qui a un objet et un chemin qui n'en as pas, le fantôme va prendre le chemin où l'objet est, sauf si l'objet est un glouton.

## B. Principaux algorithme et complexité

L'algorithme du glouton a pour objectif de réaliser étape par étape un choix optimal local, pour à terme obtenir un résultat optimal. Il privilégie la prise de décision immédiate la plus avantageuse sans se préoccuper des conséquences à long terme. Ce genre d'algorithme peut être utilisé pour la recherche de chemins.

Un autre algorithme sera utilisé, le decision tree pruning, Le tree pruning est une technique cruciale dans la construction d'arbres de décision utilisés pour les ia. Le pruning peut se produire de deux manières : le pre-pruning, qui arrête la croissance de l'arbre lorsque l'ajout de nouveaux nœuds ne contribue pas à augmenter les performances, et le post-pruning, qui intervient après la construction de l'arbre en évaluant l'importance de chaque nœud. Le processus continue jusqu'à ce qu'un critère l'arrête , basé sur la profondeur de l'arbre ou d'autres paramètres définis par le programmeur soit atteint. Le tree pruning vise à prévenir les

calculs inutiles, à simplifier, à améliorer le temps d'exécutions et à rendre l'arbre de décision plus interprétable. Cette méthode se rapproche grandement du pruning Alpha-beta pour l'algorithme du minimax qui était prévu comme algorithme au départ.

La complexité d'un algorithme glouton dépend du problème spécifique qu'il résout. En général, les algorithmes gloutons sont connus pour avoir une complexité relativement basse.  $O(n \log n)$  lorsque l'algorithme glouton impliquent un tri préalable,  $O(n)$  sinon.

## C. Ce qui est fait et non fait

Lors de cette SAÉ nous avons réussi à faire: l'algorithme du glouton expliqué ci dessus, nous avons aussi compléter toutes les API qui devait être effectué au préalable ce qui permet de pouvoir développer une IA complète. La stratégie de notre IA a été implémentée.

Au départ nous avons décidé d'implémenter l'algorithme du minimax et donc le tree pruning, mais il n'a pas été fait car l'algorithme du glouton était bien plus intéressant à effectuer pour le problème rencontré. Mais l'algorithme de tree pruning est étroitement liée au glouton, on décide de choisir la solution la plus efficace et rapidement sans se soucier des autres.