

Documentation A*

Groupe 3 Bleu : Sokoban

Goron Nathan, De La Rosa Louis-David, Basset Emilien, Demé Quentin

1 Introduction , principe :

L'algorithme A* est un algorithme dit de "Pathfinding" , il sert a déterminer , entre deux points A et B le chemin le plus adapté(le plus court ou le plus rapide dans la majorité des cas). Il existe d'autres algorithmes de recherche de chemin tels que Dijkstra ou encore l'algorithme glouton , cependant A* est , dans notre cas d'application , le plus efficace pour trouver un chemin rapidement : ce qui s'avérera très utile puisque la fonction de résolution automatique nécessitera probablement d'utiliser plusieurs fois l'algorithme par niveau.

2 Fonctionnement , implémentation dans notre projet :

Pour utiliser l'algorithme , il faudra travailler sur un environnement sous forme de graphe(déjà le cas dans notre projet) .Les cases du niveau seront représentées par des nœuds que nous classerons dans deux listes : la liste ouverte qui contiendra les nœuds a analyser et la liste fermée qui contiendra les nœuds déjà analysés

Le fonctionnement de l'algorithme se découpe ensuite en 4 parties :

- 1) A partir du point de départ (Le personnage dans notre cas) on regarde les nœuds libres parmi celui de droite , haut et bas par rapport au personnage , on place ensuite le nœud de départ dans la liste fermée.
- 2) Ensuite , pour chaque nœud voisin on distingue plusieurs cas : Si c'est un mur , on ne traite pas le nœud (il n'est ajouté dans aucune liste) , si c'est nœud déjà présent dans la liste fermée on ne le traite pas non plus(inutile de traiter deux fois le nœud. Si le nœud est déjà présent dans la liste ouverte , on vérifie sa qualité (qualité = distance entre le nœud et l'objectif). Si la qualité est meilleure on met a jour celle ci dans la liste ouverte et on met a jour le trajet final.
- 3) Après avoir traité tout les nœuds on parcourt la liste ouverte en recherchant le nœud ayant la meilleure qualité : Si la liste ouverte est vide , il n'y a pas de chemin possible entre le point A et le point B , l'algorithme s'arrête ici. Sinon , on retire le meilleur nœud de la liste ouverte pour l'insérer dans la liste fermée.
- 4)Ensuite on récupère le nœud dont la qualité est la meilleure , si celui ci est se trouve sur le point B (point d'arrivée) c'est que l'on a trouvé le chemin optimal , si ce n'est pas le cas on réitère 2)