

PATHFINDING

START

SELECT YOUR CHARACTER



Dans les jeux vidéo

le pathfinding est **omniprésent** et on l'utilise depuis Pacman. On s'en sert pour **déplacer** des personnages ou des créatures dans un **environnement** 2D ou 3D et ce dans **tout type** de jeu.



NOW YOU'RE LOST

INTRODUCTION

Le **pathfinding**, ou "recherche de chemin", se rapporte à l'**intelligence artificielle** et consiste à trouver **comment se déplacer** de manière **efficace** entre un **point de départ** et un **point d'arrivée**, éventuellement en prenant en compte certaines **contraintes**.



Très souvent, on **divise l'espace** en cases carrées, ce qui permet de **simplifier** le un tableau à **deux dimensions**. Certaines cases peuvent être **traversées**, d'autres non. On parle souvent de **noeuds** au lieu de cases lorsqu'on utilise l'approche de la **théorie des graphes**.

Algorithme de Dijkstra – 1959

Cet algorithme s'appuie sur la **théorie des graphes**. Principe : Partant du premier nœud, on compare la **distance aux autres nœuds** liés à ce nœud de départ. On choisit celui comportant la distance (ou poids) la **plus courte**. Chaque nœud choisi ouvre la voie vers d'autres nœuds. On compare alors pour chacun de ces nœuds la distance depuis le nœud de départ. On choisit le nœud qui nécessite la plus courte distance depuis le premier nœud. Ce nœud ouvre la voie à d'autres nœuds. On continue en choisissant à chaque fois la plus courte distance entre chaque nœud découvert et le nœud de départ. Lorsque l'on atteint le nœud d'arrivée, on a trouvé le **chemin optimal**.

GPS Utilisant la chimie:

Récemment, des chercheurs ont créé un système de **pathfinding** utilisant un **procédé chimique** : On dépose de l'acide à la sortie, et à l'entrée un produit qui trouve le meilleur chemin grâce à la **variation d'acidité**.

GPS

Le pathfinding trouve une **application** dans la **géolocalisation** et plus particulièrement dans la recherche d'**itinéraire**. En plus de la distance, le programme doit prendre en compte les **limitations de vitesse** (route de campagne ou autoroute) et les **conditions de circulation**.

Dans la vraie vie

Il est employé de multiples façons, dans la vie de tous les jours, et sans vraiment que nous soyons conscients de cela. L'un des exemples les plus courants est l'**intelligence artificielle**. Qu'elle soit elle-même utilisée en robotique, ou dans un jeu vidéo, une intelligence artificielle qui doit se déplacer aura toujours recours au pathfinding... à moins d'avancer dans diverses directions au hasard, évidemment.

Algorithmes

Les deux principaux algorithmes de pathfinding sont A* (prononcez "A star"), et **Dijkstra**.

A* a une complexité inférieure à celle de **Dijkstra** mais ne garantit pas de trouver le chemin optimal, contrairement à ce dernier.

Avantages/Inconvénients

Dijkstra, contrairement à **A***, trouvera systématiquement le **chemin optimal**. Cependant il est plus lent et sa complexité augmente beaucoup quand le graphe devient grand. **A*** est donc plus utile si on a besoin de **rapidité**. --> Compromis entre pertinence et rapidité.

Algorithme A* - 1968

On parcourt les cases **adjacentes** de la première case. On stocke dans une file d'attente « **open list** » les cases où l'on peut aller. Lorsque l'on stocke une case, on stocke également **son poids** (distance à la case de départ) ainsi que **son parent** (case précédant depuis laquelle on est allée sur cette case). Puis on recommence en prenant la case suivante de la liste d'attente. On s'arrête lorsqu'on a atteint la **case d'arrivée**. On remonte ensuite case par case en utilisant le **parent** de chaque case pour **reconstruire** le chemin. Si on parcourt toutes les cases de la file d'attente **sans atteindre** la case d'arrivée, alors **il n'existe pas** de chemin.

END