

TP D'INFORMATIQUE 15

Pathfinding dans une grille

1 Algorithme A*

L'algorithme A^* est une variante de l'algorithme de Dijkstra se concentrant sur un sommet destination d particulier, en se basant sur une heuristique permettant d'estimer la distance entre un sommet u et ce sommet d .

Un exemple d'heuristique est de calculer la distance euclidienne entre deux sommets, en supposant qu'ils ont une position dans le plan. Dans cette partie, on supposera qu'on a un tableau `pos` associant à chaque sommet un couple de coordonnées cartésiennes.

```
def h(u,d):
    x1,y1 = pos[u]
    x2,y2 = pos[d]
    return ((x2 - x1)**2 + (y2 - y1)**2)**0.5
```

Un autre exemple est l'heuristique nulle, qui renvoie toujours 0, et avec laquelle l'algorithme A^* se comporte comme Dijkstra.

Le pseudo-code de A^* est :

```
A_etoile(G,s,d,h):
    D[s] vaut 0
    D[u] est infinie pour les autres sommets u
    les sommets n'ont initialement pas de père
    seul le sommet s est non marqué
    tant qu'il reste un sommet non marqué :
        soit u non marqué de (D[u] + h(u,d)) minimal
        on marque u
        si u = d :
            on s'arrête en renvoyant D[d] et P
        sinon, pour chaque successeur v de u:
            on relâche (u,v)
            si on a diminué la distance de v et que v est marqué :
                on démarque v
    si on sort de la boucle principale, d n'est pas accessible depuis s
```

1. Implémenter cet algorithme en Python.
2. Écrire une fonction `chemin` prenant en argument le tableau des pères et un sommet u , et renvoyant le chemin du sommet source à u , sous forme de liste.
3. Tester avec le graphe
 $G_1 = [(1,10), (4,5)], [(2,1), (4,2)], [(3,4)], [(2,6), (0,7)], [(1,3), (2,9), (3,2)],$
les positions `pos = [(0, 0), (0, 1), (0, 2), (1, 2), (1, 1)]`, et vérifier que l'heuristique nulle et l'heuristique euclidienne donnent le même chemin.
4. Si l'heuristique surestime la distance réelle de u à d , l'algorithme A^* peut renvoyer un chemin non optimal. Observer ce phénomène en modifiant les positions des sommets.

2 Application au pathfinding

Dans beaucoup de jeux vidéos, il est nécessaire de calculer rapidement la trajectoire que doit choisir une entité dans un environnement (*pathfinding*). L'objectif de cette partie est d'observer que l'heuristique de A^* permet de jouer sur ce compromis entre vitesse de calcul et qualité du chemin renvoyé.

