

ALGORITHME A*_(star)

ou dit de

PATHFINDER

L'algorithme A^* ou dit de PathFinder, est un algorithme simple permettant de trouver le plus court chemin d'un point A au point B en évitant les obstacles.

Très utilisé dans les jeux de stratégie.

A^* est un algorithme heuristique de recherche du plus court chemin (le plus souvent possible).

Le fonctionnement de l'algorithme A*

Comme première action, l'algorithme A* ajoute la case de départ à la liste ouverte.

Il y a deux listes utilisées par A*.

Ces listes contiennent des nodes (des cases). Les deux listes sont:

La liste ouverte, elle contient les nodes à analyser.

La liste fermée, elle contient les nodes déjà analysés.

L'algorithme A^* commence par étudier la case de départ puisqu'elle est la seule dans la liste ouverte.

Il ajoute à la liste ouverte toutes les cases adjacentes en indiquant leur parent et leur poids.

Cette phrase contient de nombreux mots-clés que l'on va voir.

Premièrement, « leur parent », c'est la case de départ.

Cette notion de parent est très importante car c'est elle qui va nous permettre de retrouver le chemin à partir de la case d'arrivée.

Deuxièmement, le poids, on le note F .

Il est la somme de la distance parcourue pour arriver à ce point depuis le point A (G) et de celle à "vol d'oiseau" restant jusqu'au point B (H).

Nous utiliserons en général, pour les distances 10 pour celle d'une case à une autre en ligne droite, 14 en diagonale.

Pour la distance à vol d'oiseau, nous utiliserons la distance de Manhattan. Cette distance correspond à la distance entre deux points en abscisse plus celle en ordonnée.

Ensuite, on enlève la case de départ de la liste ouverte pour la mettre dans la liste fermée.

Le principe de base est de se dire que pour chaque case, on peut calculer un score, et que plus ce score est bas, plus la case a la de chance d'être sur le meilleur chemin. Toute la difficulté vient de la façon de calculer ce fameux score.

Pour calculer ce score, on va utiliser deux choses :

- Un cout et une heuristique.
- Dans des versions simples, le cout n'est que le nombre de cases parcourues depuis la case de départ.
- Pour corser les choses, on peut ajouter d'autres valeurs dans ce cout, comme le type de terrain.
(Par exemple béton = +0, herbe = +1, boue = +10).

L'heuristique, c'est une règle sans fondement (c'est-à-dire qui marche mais sans véritable raison) qu'on fixe et qui permet de juger de l'importance d'une position par rapport au chemin à rechercher. La plupart des algorithmes d'A utilisent soit la distance euclidienne entre la case courante et la destination :

$$\sqrt{((PC.x - PO.x)^2 + (PC.y - PO.y)^2)}$$

- Soit, plus rapide à calculer, le Manhattan, qui est une approximation grosso modo valable dont la formule est

$$(PC.x - PO.x) + (PC.y - PO.y)$$

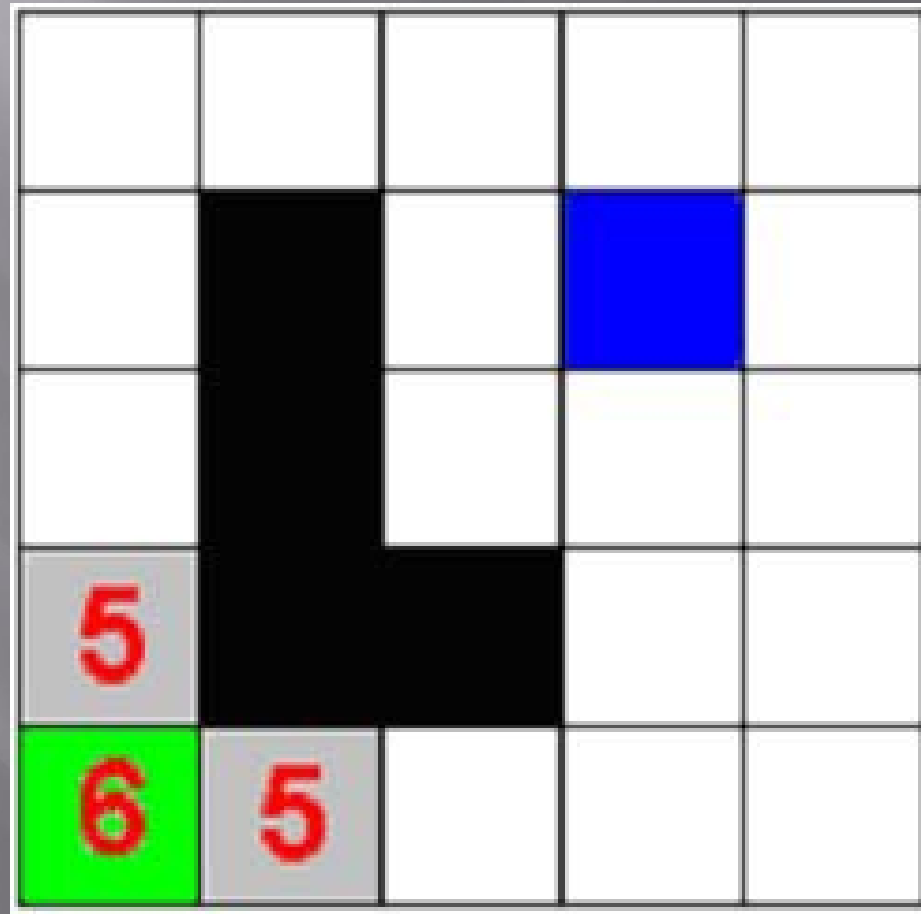
Une fois qu'on a défini ces deux valeurs, on parcourt la carte en partant du point de départ, on calcule le score de toutes les destinations possibles, puis on recommence sur toutes les destinations, en commençant par celles qui ont le plus petit score. ..

Voyons maintenant ce que ça donne sur un exemple simple.

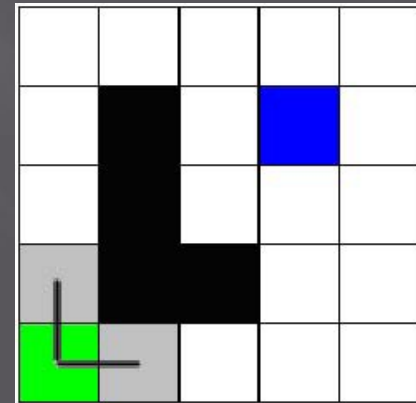
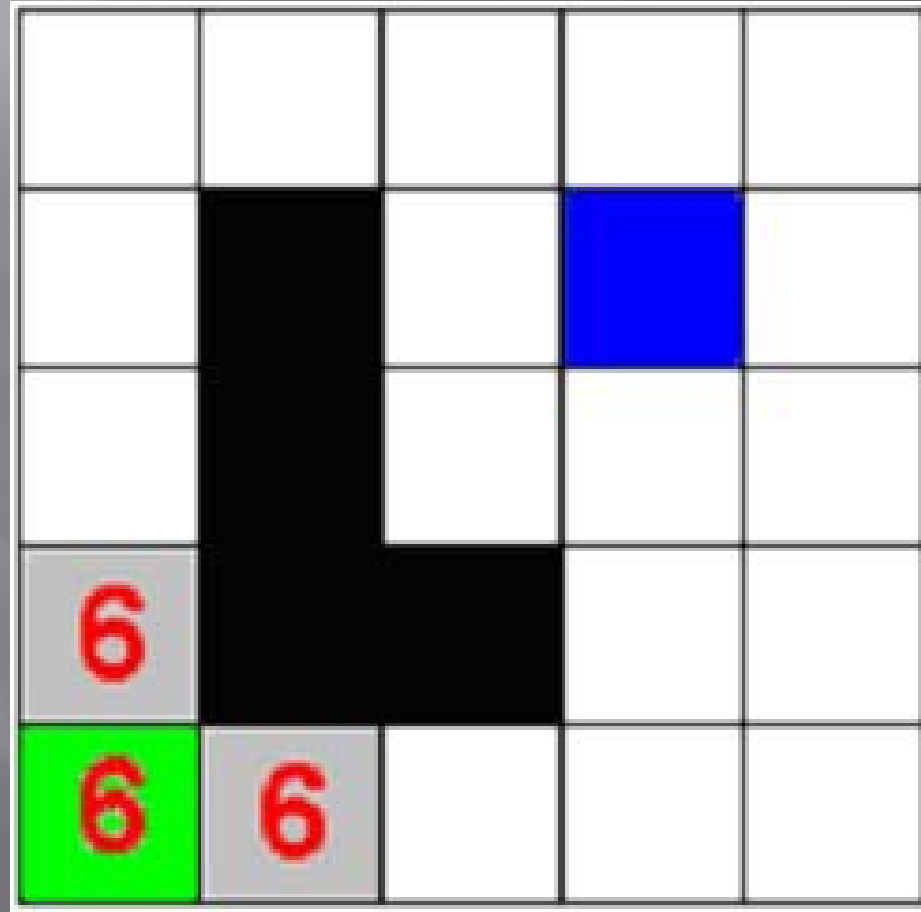
NB : Dans notre exemple, chaque déplacement d'une case vaut 1 en ligne droite et en diagonale pour simplifier la compréhension de nos calculs.

Etape 1

Etape 1 - valeur de H

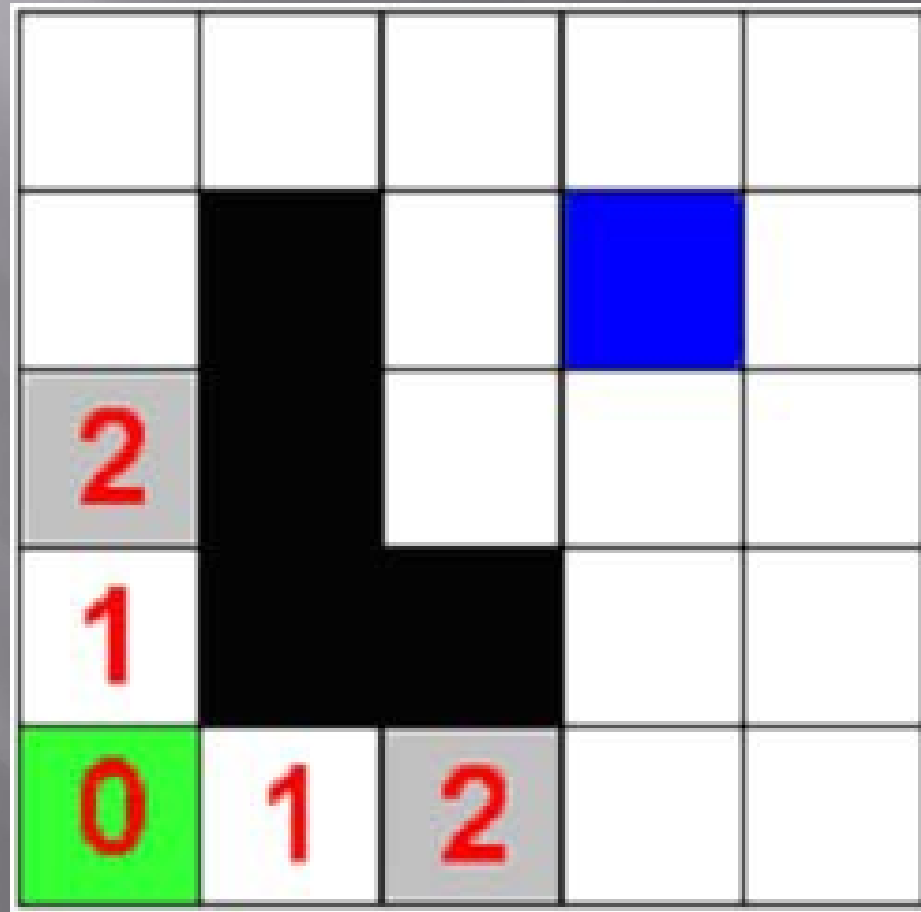


Etape 1 - valeur de F

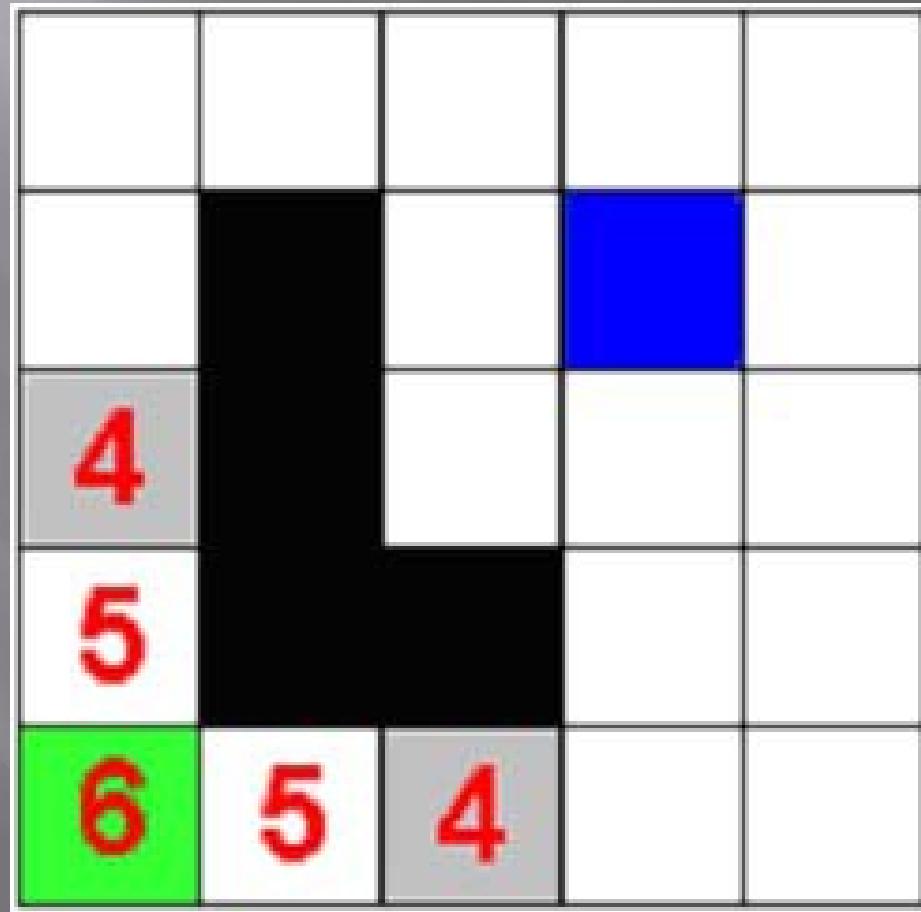


Etape 2

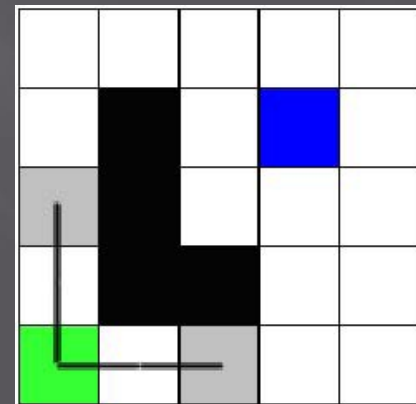
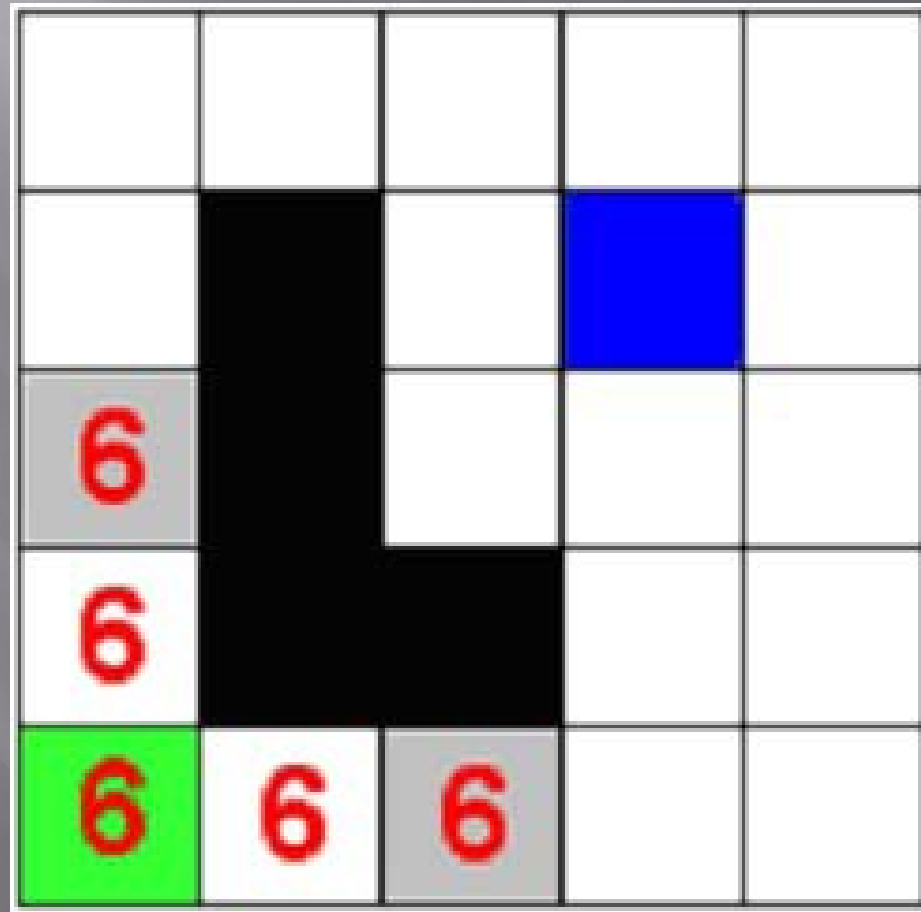
Etape 2 - valeur de G



Etape 2 - valeur de H



Etape 2 - valeur de F



Etape 3

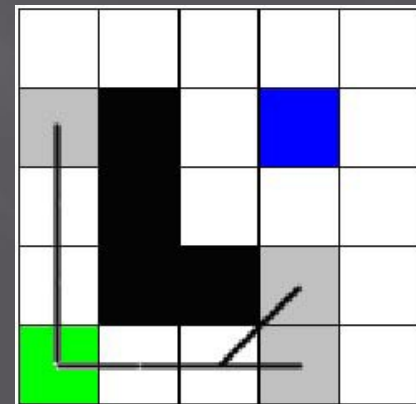
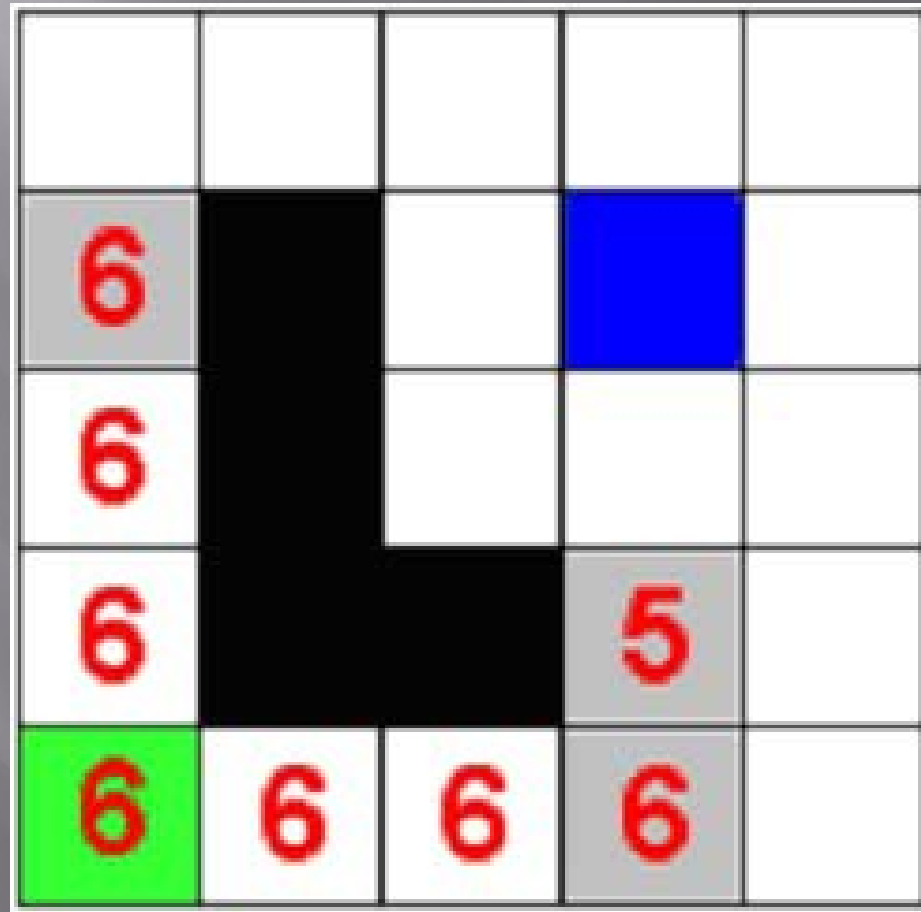
Etape 3 - valeur de G

3				
2				
1			3	
0	1	2	3	

Etape 3 - valeur de H

3				
4				
5			2	
6	5	4	3	

Etape 3 - valeur de F



Etape 4

Etape 4 - valeur de G

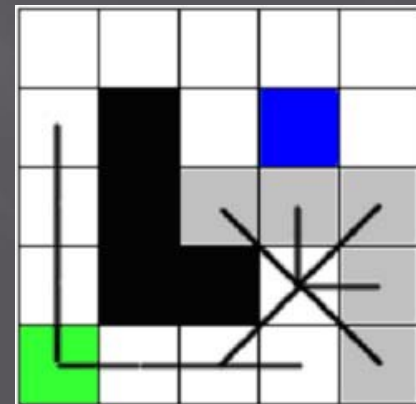
3				
2		4	4	4
1			3	4
0	1	2	3	4

Etape 4 - valeur de H

3				
4		2	1	2
5			2	3
6	5	4	3	4

Etape 4 - valeur de F

6				
6		6	5	6
6			5	7
6	6	6	6	8



Etape 5

Etape 5 - valeur de G

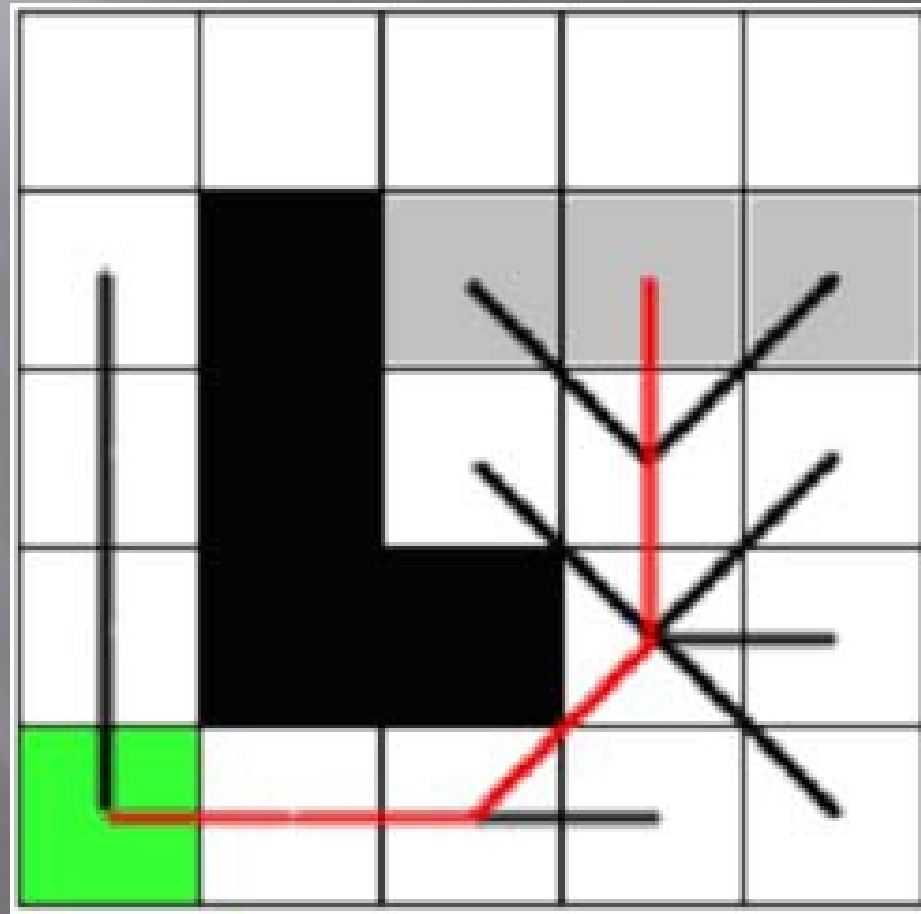
3		5	5	5
2		4	4	4
1			3	4
0	1	2	3	4

Etape 5 - valeur de H

3		1	0	1
4		2	1	2
5			2	3
6	5	4	3	4

Etape Finale

Il ne reste plus qu'à remonter la liste chaînée arrière
Pour trouver le chemin.



Implémentation de l'algorithme

1) Ajouter la case de départ à la liste ouverte.

2) Répéter ce qui va suivre :

a) repérer la case avec le poids F le plus petit dans la liste ouverte.
Ce sera la case actuelle.

b) la mettre dans la liste fermée.

c) pour chacune des huit cases adjacentes à la case actuelle...

* Si la case est un obstacle ou est dans la liste fermée, ignorez-la. Sinon, faites ce qui suit.

* Si la case n'est pas dans la liste ouverte, ajoutez-la. Définissez la case actuelle comme parent de cette case. Enregistrez le F, le G et le H de la case.

* Si elle est déjà sur la liste ouverte, vérifiez si un nouveau G sera plus faible. S'il est plus faible, changez le parent de cette case pour la case courante, et recalculez les scores F et G.

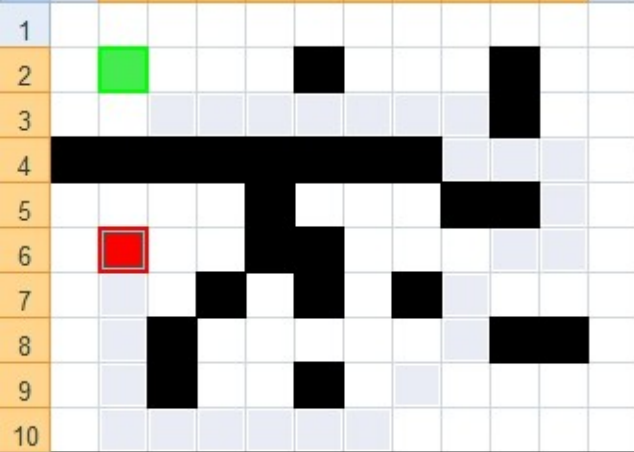
d) arrêtez-vous si :

* vous ajoutez la case d'arrivée à la liste fermée,

* ou si la liste ouverte est vide (dans ce cas, il n'y a pas de solution).

3) Reconstituez le chemin à partir des parents (chainage arrière).

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38



cell. explorées	89
cell. utilisées	27
efficacité	30%
Coût estimé	40
Coût effectif	276
difficulté	6,9

Pathfinder

Score F

13	68	60	68	82	96	110	124	144	164	184	204	224
14	54	40	54	68	82		130	144	164		210	224
15	48	40	48	62	76	96	116	136	156		216	230
16								162	182	202	222	
17					222	222	228			208	222	
18	290	276				214	214	214	214	214	228	
19	284	276		242		228		222	222	228	242	
20	284	276		236	236	236	236	236	236		256	
21	284	276		250	250		244	244	250	270	270	270
22	290	276	270	264	258	252	252	258	264	278	278	284

Score G

14	10	14	24	34	44	54	64	74	84	94	104	
10	0	10	20	30		64	68	78		104	108	
14	10	14	24	34	44	54	64	74		114	118	
								84	94	104	114	
					178	168	164			114	118	
280	276					164	154	144	134	124	128	
270	266			208		174		148	138	134	138	
260	256			208	198	188	178	168	158		148	
250	246			212	208		182	172	168	178	168	158
246	236	226	216	206	196	186	182	178	182	172	168	

Score H

54	50	54	58	62	66	70	80	90	100	110	120	
44	40	44	48	52		66	76	86		106	116	
34	30	34	38	42	52	62	72	82		102	112	
								78	88	98	108	
					44	54	64			94	104	
10	0					50	60	70	80	90	100	
14	10			34		54		74	84	94	104	
24	20			28	38	48	58	68	78		108	
34	30			38	42		62	72	82	92	102	112
44	40	44	48	52	56	66	76	86	96	106	116	