



Projet d'Informatique Scientifique

Etudiante: Elizabeth Gandibleux

Groupe: 681C

Encadrant : **Prof. Dr. Hab. Xavier Gandibleux**

1 Partie 1

1.1 Execution du code

Les instances *arena.map* et *didactic.map* ont été choisies pour tester le programme. Par défaut, le point de départ est: **(5,4)** et le point d'arrivée est **(10,7)**.

Dans un premier temps il faudra sélectionner l'instance voulue

Instances numériques à choisir:

```
arena1.map  
> didactic.map
```

Ici nous sélectionnons *didactic.map*.

Dans un second temps, il faudra choisir l'algorithme qui est choisit pour être exécuté ou alors demander au programme d'exécuté tous les algorithmes

Algorithme(s) à choisir:

```
Dijkstra  
A*  
> Tous
```

1.2 Approche "recherche informée optimale": *Algorithme de dijkstra*

1.2.1 Structures choisies

- Matrice de caracteres: **A**
- Matrice d'entiers: **matriceNumerique**
- Matrice de tuples d'entiers: **matriceOriginelle**
- Une file de priorité, autrement dit un tas: **pQ**
- File de tuples d'entiers: **listeFinale**
- Booléen : **trouve**

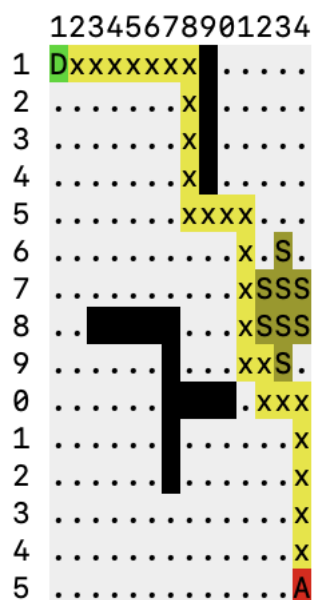
1.2.2 Jeux de test

Instance choisie: didactic.map

D::Tuple{Int64,Int64} = (1,1)

Ar::Tuple{Int64,Int64} = (15,14)

Dijkstra Algorithmme ...



Distance D → A: 27

Number of states evaluated: 193

Dijkstra fait ...

1.3 Approche "recherche informée heuristique": *Algorithme du A**

1.3.1 Structures choisies

Les structures choisies sont les mêmes que celles dans l'algorithme de Dijkstra

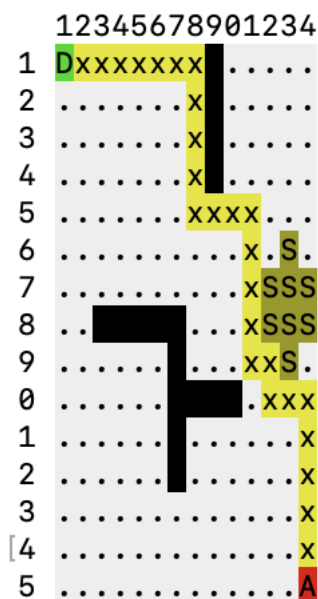
Instance choisie: didactic.map

D::Tuple{Int64,Int64} = (1,1)

Ar::Tuple{Int64,Int64} = (15,14)

1.3.2 Jeux de test

A* Algorithmme ...



Distance D → A: 27

Number of states evaluated: 173

A* fait ...

2 Partie 2

2.1 Formules

L'algorithme **WA*** est une extension de **A***. En effet, cela se voit notamment en observant les formules de ces deux codes:

WA*

$$f(n) = (1-w) \cdot g(n) + w \cdot h(n)$$

Lorsque $w = 1$:

A*

$$f(n) = h(n)$$

Lorsque $w = 0$:

Di

$$f(n) = g(n)$$

qui est la formule de l'algorithme **Dijkstra**

2.2 Analyse

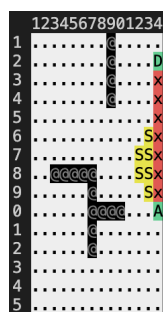
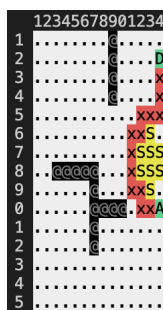
En résumé, l'algorithme **WA*** se base sur le même principe que le **A star** sauf que la fonction h est multipliée par un facteur w , un certain poids.

Après exécution, on voit que le premier code est certes plus *performant* et visite moins d'états, mais il est nettement moins *précis* que le second.

2.3 Situations

Voici différentes situations didactique, facile et difficile (dans l'ordre):

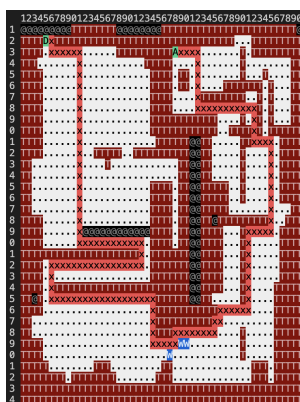
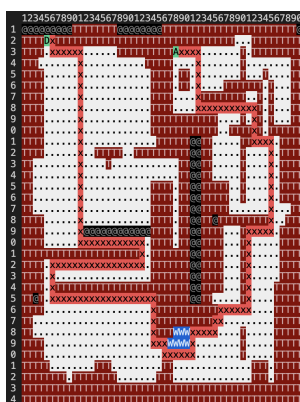
2.3.1 didactic.map



2.3.2 arena.map



2.3.3 den009d.map



Dans ces 3 cas on voit très facilement que les deux algorithmes A^* et WA^* réagissent différemment. Ce qui s'explique par le fait que l'on donne plus ou moins d'importance à la partie gloutonne à l'aide w .

3 Conclusion

En conclusion, parmi les 3 algorithmes étudiés **Dijkstra** est le plus précis tandis que le **WA*** est le moins. Ce qui est l'inverse pour le nombres d'états visités, toujours avec **A*** qui reste au milieu. Enfin, en terme de rapidité du code, les 3 sont rapides d'exécution.

4 Plus encore...

D'autres algorithmes peuvent aussi avoir leur place dans un sujet de recherche plus poussé, comme l'**Improved Optimistic Search**, l' **Optimistic search** et le **Dynamic potential search**.