

Rapport Circuit 3

Comment installer et utiliser notre programme

- Se déplacer dans le dossier src grâce à votre terminal.
- Lancer javac Controleur/*.java
- Lancer java Controleur.Launcher

Problèmes rencontrés et les choix réalisés

- Sérialisation (pour pouvoir jouer avec des robots) : problématique pour les circuits avec bonus car ils sont dispersés aléatoirement. Ceci n'était donc pas adapté et nous avons donc revu totalement l'implémentation de trajets aléatoires.
- Circuit forme "infini" : impossible à faire pour des algorithmes notamment A*, concernant l'implémentation dans une partie, cela aurait nécessité beaucoup trop de changement, et vu le temps restant, nous avons donc préféré laisser tomber l'idée.
- Bonus/Malus: 1ère version implémentée sur les sommets comme dans le sujet. Le problème de cette version est qu'elle augmentait le nombre de possibilités, cela n'était pas compatible avec l'affichage qui ne peut afficher que jusqu'à 9 possibilités.
2ème version pour le joueur uniquement. Cette version traitait le problème des possibilités mais était à moitié fonctionnelle car elle ne fonctionnait que sur un tour et n'était pas applicable aux algorithmes.
Version finale fonctionnelle , appliquée sur les sommets pour être utilisable partout grâce aux attributs dans sommet et des fonctions dans les différentes classes permettant de transmettre les informations sur 3 tours.
- File de priorité : Au départ, nous avions en attribut, un tableau de Noeud et un tableau d'int pour la variable IndiceDansTas. Ce n'était évidemment pas une bonne solution car nous avons besoin d'une Hashmap pour réellement représenter IndiceDansTas et d'une ArrayList de Noeud afin de ne pas avoir à se soucier de la taille du tableau.
- Graphique : Interface graphique incompatible avec Macos (l'affichage ne se fait pas correctement). Nous avons essayé d'installer une machine virtuelle pour Mac et ainsi pouvoir adapter le code sauf qu' aucune machine ne

fonctionnait. Nous n'avons donc qu'une interface graphique fonctionnelle uniquement pour Windows et Ubuntu.

- Trajet aléatoire : Après la sérialisation, nous avons implémenté tout un processus pour obtenir des trajets aléatoires. Nous avons tout d'abord dupliqué des algorithmes pour finalement créer des fonctions auxiliaires utilisant les algorithmes de base. Après certains tests, nous avons remarqué un problème survenant uniquement sur le circuit 4 : lors d'une partie avec des robots, il est possible qu'un chemin ne soit pas trouvé. Après une longue discussion commune, nous avons conclu que sachant que le circuit 4 est un circuit supplémentaire, il était préférable de ne pas modifier notre code (car cela rendrait nos trajets aléatoire moins pertinents : trajets proches voire égaux au PCC), car cela fonctionnait sur les circuits principaux et donc de garder cette erreur. La modification qui aurait pu être effectuée pour pallier ce problème (et la seule qui nous paraissait appropriée) est l'ajout d'un algorithme de Dijkstra prenant les 3 premiers chemins qui finissent et les retournant pour les utiliser.

Les différents algorithmes implémentés

Nous avons implémenté 3 algorithmes : Parcours en Largeur, Dijkstra et A*. En plus de cela, nous les avons modifiés pour implémenter une variante (on arrive sur l'arrivée avec une vitesse nulle). Nous avons donc en plus 3 variantes (une pour chaque algorithme).

Lors de nos nombreux tests nous avons remarqué que les algorithmes de parcours en largeur et de Dijkstra sont quasiment identiques au niveau de leurs performances pour n'importe quel circuit. Ils sont tous les deux très performants mais peuvent prendre plus de temps et de mémoire proportionnellement au nombre d'espaces n'étant pas un mur.

L'algorithme A*, est moins performant sur les circuits étant composés d'une majorité de murs. Sur le circuit 5, qui est presque dénué de murs, A* performe aussi bien que Dijkstra (voir Image Comparaison A* et Dijkstra). En les lançants séparément, on peut remarquer qu'A* se lance quasiment instantanément tandis que Dijkstra se lance après un peu moins de 25 secondes. Cela montre qu'A*, malgré son manque de performance sur certains circuits, il semblerait plus pertinent à utiliser dans certains cas.

Répartition du travail

Benidy MIKIA :

- Parseur PPM, pour que l'on puisse lire le fichier et le transformer en tableau de pixel(Ugo à fait la partie pour passer d'un tableau de pixel en tableau de int) +fonction pour récupérer le départ et l'arrivée
- Algorithme de parcours en largeur
- Fonction Allpossibilite() avec Pauline
- Début de la fonction trajet sans mur (complétée et corrigée par Pauline et Ugo)
- implémentation de Bonus/Malus

Aurélie NODIN :

- Création de la classe Noeud qui regroupe un Sommet (Sommet.java) et une distance (int).
- Implémentation intégrale de la FileDePriorite.
- Création de la classe Joueur.
- Implémentation de tout ce qu'il faut pour faire une Partie Solo.
- Implémentation de tout ce qu'il faut pour faire une Partie Multijoueur Locale (maximum 3 joueurs en plus donc 4 au total).

Pauline ADAM :

- Algorithme de Dijkstra + ajout des variantes + modification des algorithmes pour retourner soit un PCC, soit un chemin avec une distance précise + fonctions AllPossibilite et trajet sans mur en collaboration avec Benidy et Ugo
- Concours entre 2 algorithmes
- Trajet aléatoire (+ ancienne sérialisation)
- Partie avec des robots (3 robots maximum)
- Création de nouveaux circuits : 2 (+ 2 de forme infini qui ne sont pas utilisés car trop complexe à implémenter)
- Partie graphique
- Programme textuel pour la console

Ugo TORIS :

- Modification du parseur afin qu'il convertisse le tableau de pixels en tableau d'entiers
- Algorithme A*
- Correction de la fonction trajet sans mur

Exemples tirés du programme

```
Bienvenue dans le Programme.  
Comment voulez vous utiliser le Programme ?  
    1 : En Mode Graphique  
    2 : En Mode Console  
    A : Revenir a l'accueil  
    E : Quitter le programme  
  
Entrez votre choix :
```

Partie graphique :



Choisissez votre Circuit



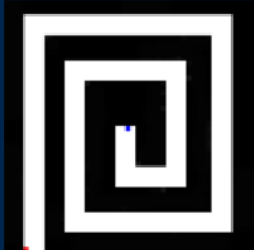
☒ Circuit 1



☐ Circuit 2



☐ Circuit 3



☐ Circuit 4



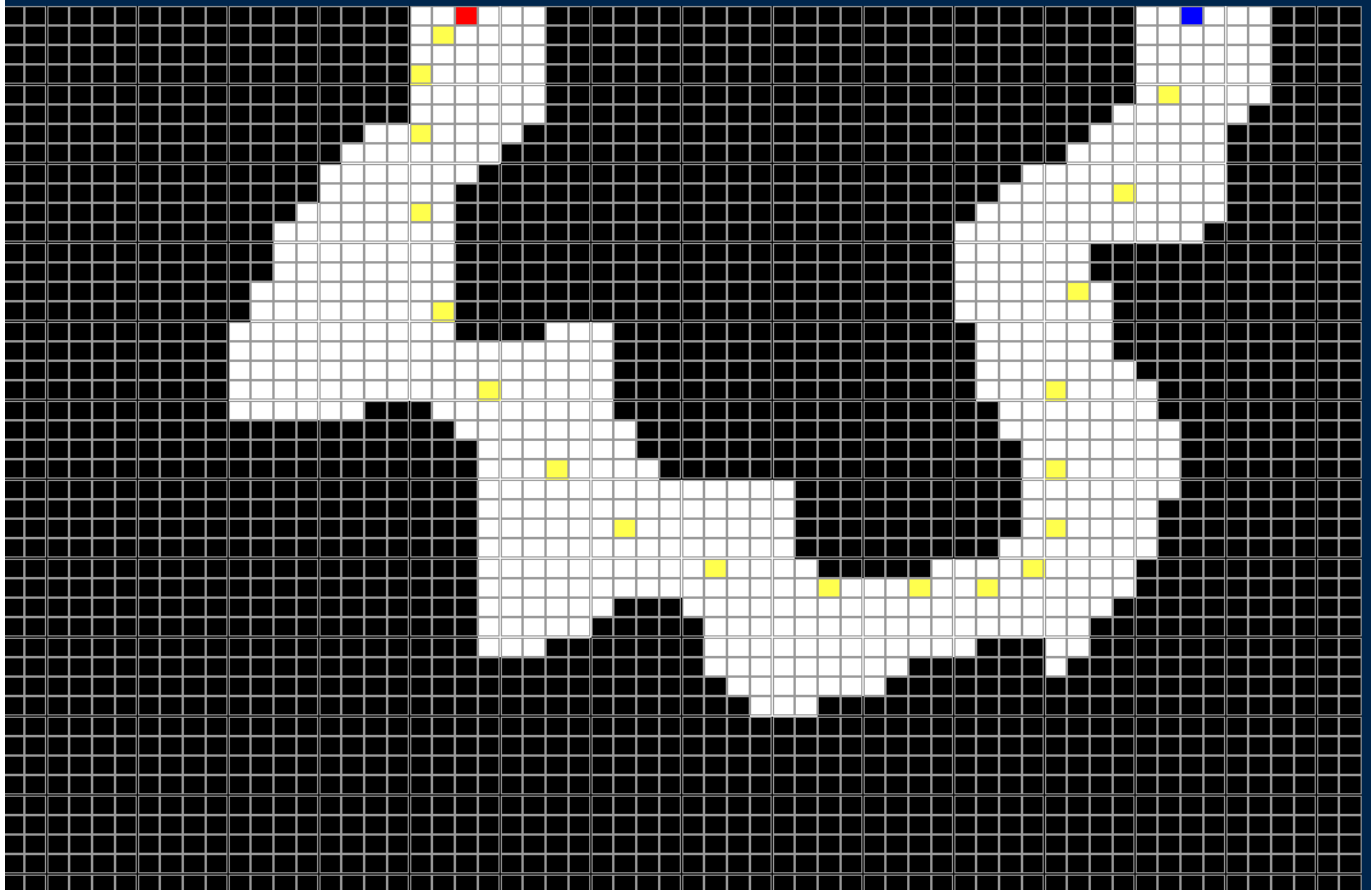
☐ Circuit 5

Voir le chemin

Retour

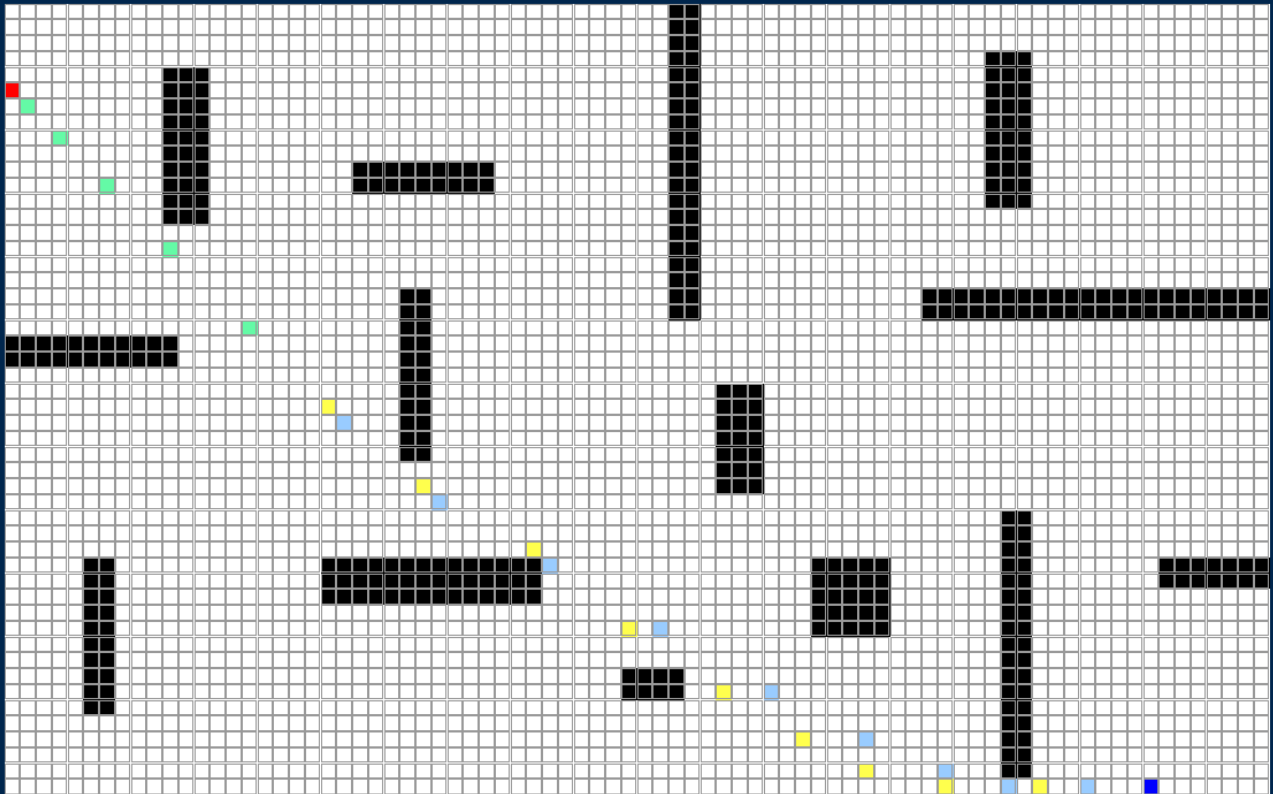
Quitter

PL, sur le circuit1, sans bonus, avec une distance de 20

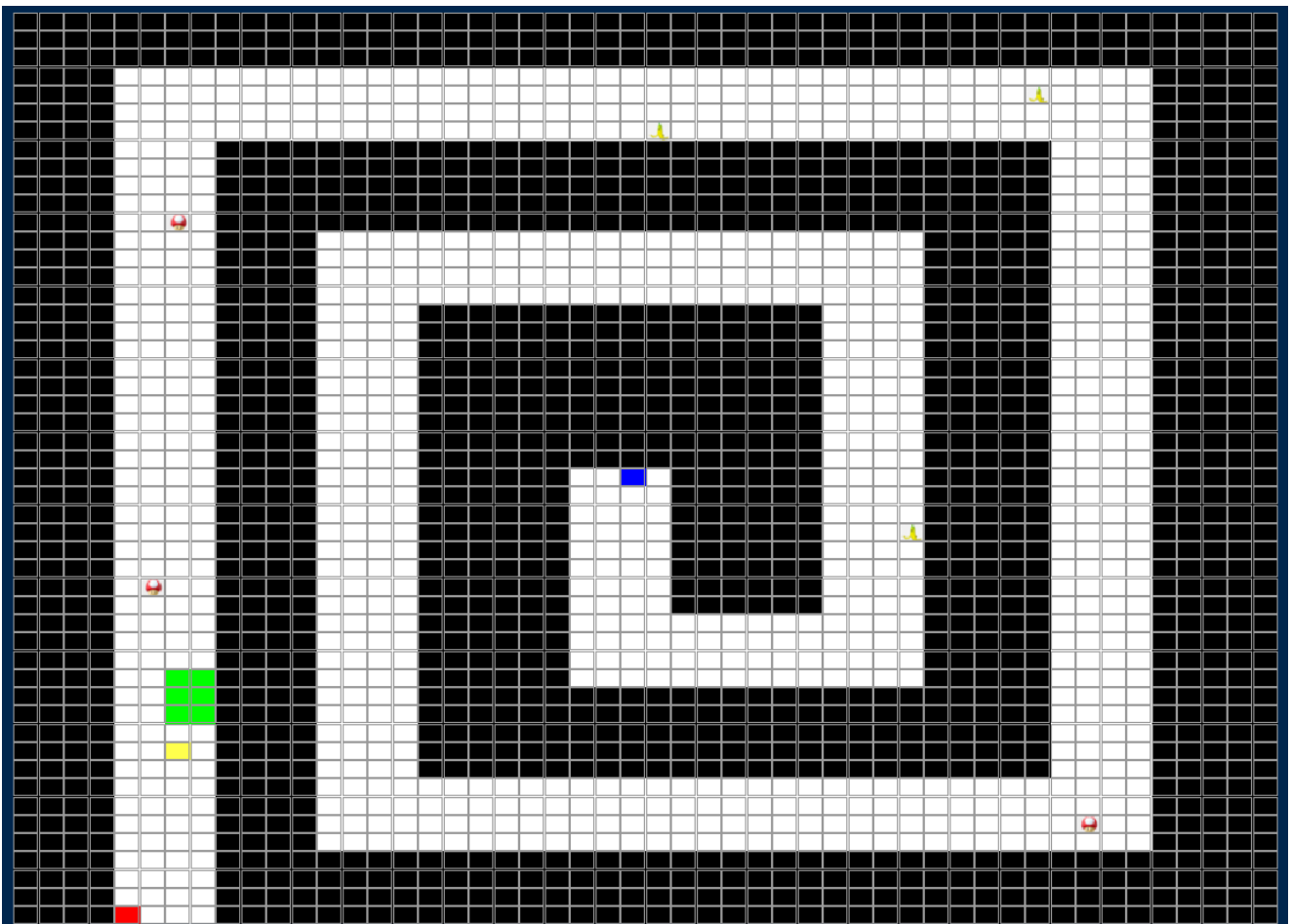


Accueil

Comparaison de PL (15) et A* (15)



Accueil



Partie Console :

Que voulez-vous faire ?

- 1 : Voir une demo (trajet Aleatoire)
- 2 : Lancer un Algorithme (PCC)
- 3 : Faire une Comparaison entre 2 Algorithmes
- 4 : Jouer sur un circuit
- E : Quitter le programme

Choisissez l'Algorithme.

- 1 : Parcours en Largeur
- 2 : Dijkstra
- 3 : A*
- 4 : Variante du PL (arrivee a vitesse nulle)
- 5 : Variante de Dijkstra (arrivee a vitesse nulle)
- 6 : Variante de A* (arrivee a vitesse nulle)
- A : Revenir a l'accueil
- E : Quitter le programme

Voulez vous implementez des bonus ?

- 1 : Oui (bonus et malus)
- 2 : Non (aucun boost)
- A : Revenir a l'accueil
- E : Quitter le programme

Choisissez un circuit :

- 1 : Circuit 1
- 2 : Circuit 2
- 3 : Circuit 3
- 4 : Circuit 4
- 5 : Circuit 5
- A : Revenir a l'accueil
- E : Quitter le programme

Votre Algorithmme 1 est : Dijkstra
Votre Algorithmme 2 est : A*

```
.....XX.....
.....XX.....
.....XX.....
.....XX.....XXX.....
....XXX.....XX.....XXX.....
A....XXX.....XX.....XXX.....
.~....XXX.....XX.....XXX.....
....XXX.....XX.....XXX.....
.~....XXX.....XX.....XXX.....
....XXX.....XX.....XXX.....
....XXX.....XXXXXXXXXX.....XX.....XXX.....
....~....XXX.....XXXXXXXXXX.....XX.....XXX.....
....XXX.....XX.....XXX.....
....~....XX.....XX.....
.....XX.....XX.....XXXXXXXXXXXXXXXXXXXXXXXXXXXX
.....XX.....XX.....XXXXXXXXXXXXXXXXXXXXXXXXXXXX
....~....XX.....
XXXXXXXXXXXXX.....XX.....
XXXXXXXXXXXXX.....XX.....
.....XX.....
.....XX.....XXX.....
.....XX.....XXX.....
.....~....XX.....XXX.....
.....XX.....XXX.....
.....XX.....XXX.....
.....XXX.....
.....~....
.....XX.....
.....XX.....
.....XX.....
....XX.....XXXXXXXXXXXXX~....XXXXX.....XX.....XXXXXX
....XX.....XXXXXXXXXXXXX.....XXXXX.....XX.....XXXXXX
....XX.....XXXXXXXXXXXXX.....XXXXX.....XX.....
....XX.....XXXXX.....XX.....
....XX.....12.....XXXXX.....XX.....
....XX.....XX.....
....XX.....XXXXX.....XX.....
....XX.....XXXXX.....1.2.....XX.....
....XX.....
.....XX.....
.....1.2.....XX.....
.....XX.....
.....1.2.....XX.....
.....1.2.1.2...B.....
```


Vous avez decide de lancer le PCC de PL sur le circuit4 avec des bonus/malus.

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXX.....o....o....o....o.....XXXX
XXXX.....o.....o.....XXXX
XXXX.....o.....XXXX
XXXX.....o...XXXX
XXXX...oXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX
XXXX...XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX
XXXX...XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..o.XXXX
XXXX.o..XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX
XXXX...XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX
XXXX...XXXX.....XXXX...XXXX
XXXX...XXXX.....o...o.....XXXXX...oXXXX
XXXXo...XXXX...o.....o.....XXXXX...XXXX
XXXX...XXXX.....o...XXXXX...XXXX
XXXX...XXXX...oXXXXXXXXXXXXXXXXXXXX...XXXXX...XXXX
XXXX...XXXX...XXXXXXXXXXXXXXXXXXXX.o..XXXXX...XXXX
XXXX...XXXX...XXXXXXXXXXXXXXXXXXXX...XXXXX...oXXXX
XXXXo...XXXX..o.XXXXXXXXXXXXXXXXXXXXX...XXXXX...XXXX
XXXX...XXXX...XXXXXXXXXXXXXXXXXXXX..o.XXXXX...XXXX
XXXX...XXXX...XXXXXXXXXXXXXXXXXXXX...XXXXX...XXXX
XXXX...XXXX...XXXXXXXXXXXXXXXXXXXX...XXXXX...XXXX
XXXX...XXXX.o..XXXXXXXXXXXXXXXXXXXX...XXXXX...XXXX
XXXX...XXXX...XXXXXXXXXXXXXXXXXXXX..o.XXXXX...oXXXX
XXXX.o..XXXX...XXXXX..B.XXXXXX...XXXXX...XXXX
XXXX..+.XXXX...XXXXX...XXXXX...XXXXX...XXXX
XXXX...XXXX...XXXXX...XXXXX...XXXXX...XXXX
XXXX...XXXX.o..XXXXX..o.XXXXXX..o.XXXXX...XXXX
XXXX-...XXXX...XXXXX...XXXXX...XXXXX...XXXX
XXXX.o..XXXX...XXXXX...XXXXX...XXXXX...oXXXX
XXXX...XXXX-...XXXXX...oXXXXX..o.XXXXX...XXXX
XXXX...XXXX...XXXXX...XXXXX+...XXXXX...XXXX
XXXX...XXXX.o..XXXXX...o.....o..XXXXX...XXXX
XXXXo...XXXX...XXXXX.....o..o...XXXXX...XXXX
XXXX...XXXX...XXXXX.....XXXXX...oXXXX
XXXX...XXXX...XXXXX.....XXXXX...XXXX
XXXX...XXXX.o..XXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX
XXXX...XXXX...XXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX
XXXXo...XXXX...XXXXXXXXXXXXXXXXXXXXXXXXXXXX..o.XXXX
XXXX...XXXX..o.XXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX
XXXX...XXXX...-XXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX
XXXX...XXXX...o.....o.....XXXXX
XXXXo...XXXX...o..o.....XXXXX
XXXX...XXXX.....o.....o.....XXXX
XXXX...XXXX.....o...o.....+.XXXX
XXXXo...XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXX...XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXo...XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXA...XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

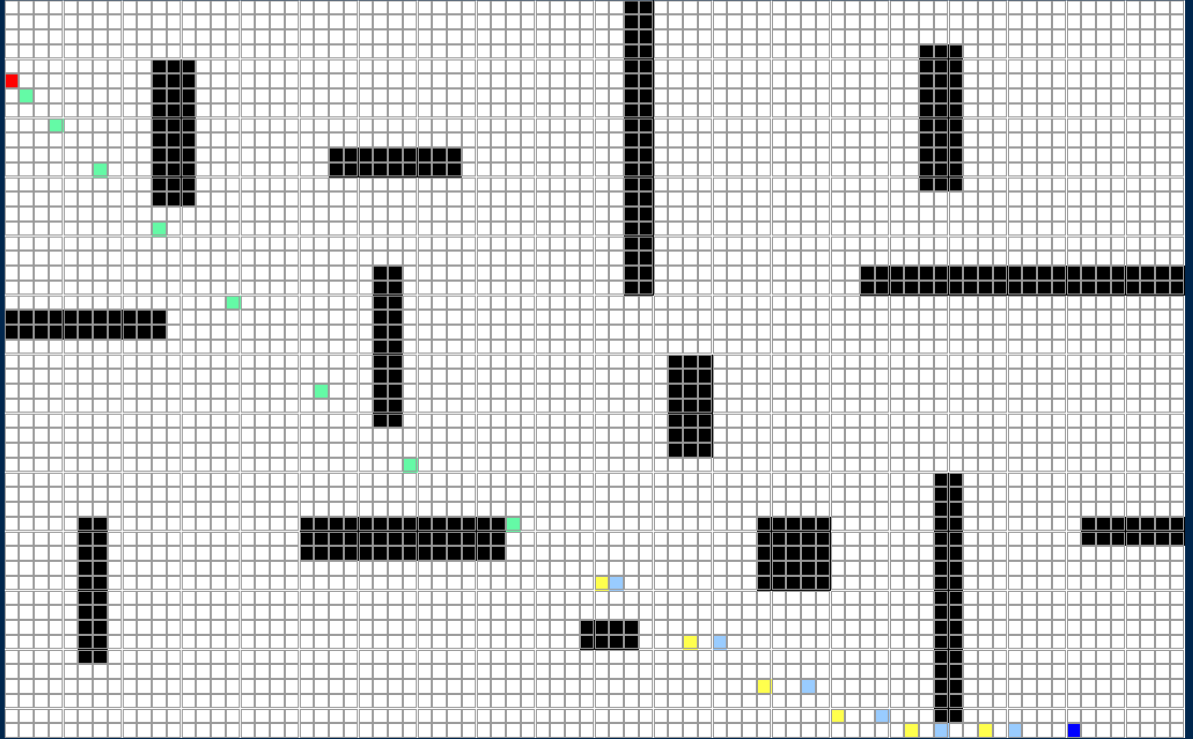
Distance parcourue :58.

Vitesse sur X :0, Vitesse sur Y :-3

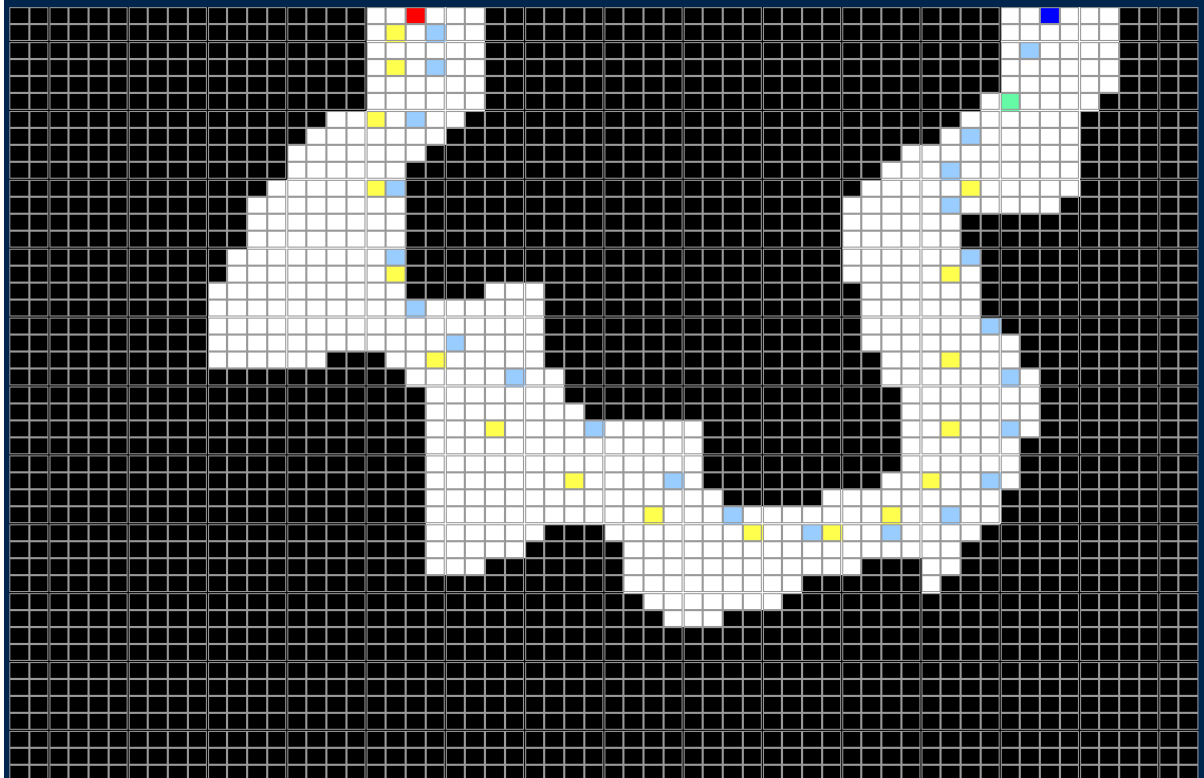
Comparaisons des algorithmes

Comparaisons A* - Dijkstra

Comparaison de Dijkstra (15) et A* (15)

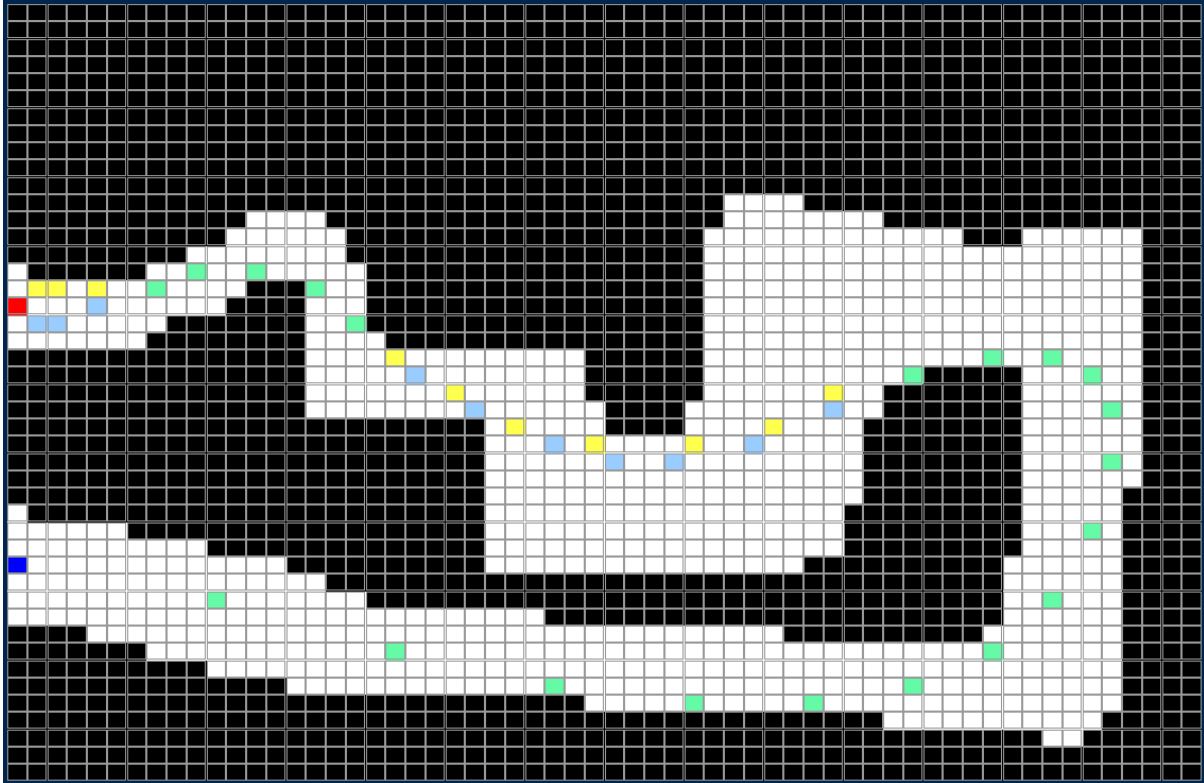


Comparaison de Dijkstra (19) et A* (25)



Comparaisons PL - Dijkstra

Comparaison de PL (31) et Dijkstra (31)



Comparaison de PL (variante) (18) et Dijkstra (variante) (18)

