

Projet Annuel

# Rapport

étape 1 : recherche de plus court chemin.



**SIMON, Pierre**  
avril 23, 2018



# CONTENU

## RECHERCHE LOCALE NAÏVE 3

Postulat 3

Valeur clé 3

## RECUIT SIMULE 4

Postulat 4

Valeur clé 4

## ALGORITHMME GENETIQUE 5

Postulat 5

Valeur clé 5

## DJIKSTRA 6







Postulat 6

## A STAR 6

Postulat 7

## DJIKSTRA VS A\* 8

# Recherche locale naïve

PROBLEME	1	2	3	4	5	6
RESULTAT						

## Postulat

**Initialisation** : Génération d'un ensemble d'action aléatoire qui forme un chemin.

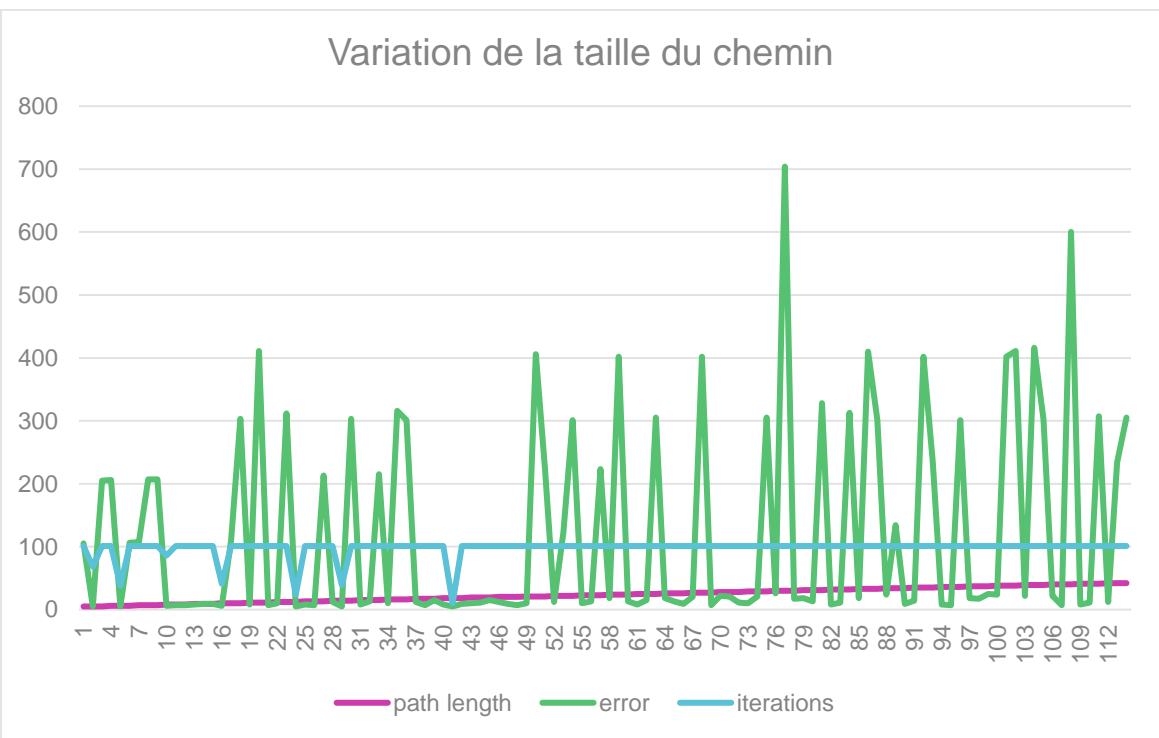
**Modification** : Une action d'indice aléatoire par une nouvelle action aléatoire.

**Critère de sélection** : Le nouveau chemin est meilleur que l'ancien.

**Critère d'arrêt** : L'erreur est égale à l'erreur minimale.







## Valeur clé

**La taille du chemin** : c'est la valeur la plus influente, un chemin trop long va avoir du mal à tendre vers la solution et réciproquement. On obtient des résultats positif à partir d'une taille de 10 et cela semble être optimal au alentours de 50. Ce qui semble logique du à la taille de la carte 50x50.



Lorsque l'itération atteint 100, cela signifie que la solution n'a pas été trouvée dans ces conditions.

## Recuit simulé

PROBLEME	1	2	3	4	5	6
RESULTAT						

### Postulat

**Initialisation** : Génération d'un ensemble d'action aléatoire qui forme un chemin.

**Modification** : Une action d'indice aléatoire par une nouvelle action aléatoire.

**Critère de sélection** : Si une valeur aléatoire entre 0f et 1f est inférieure au critère de Métropolis soit l'exponentiel de la différence entre l'erreur de l'état précédent et l'erreur de l'état courant divisé par la température alors on prend le nouveau chemin. Si l'erreur du nouveau chemin est toujours la même on incrémente la stagnation sinon on la remet à 0. Si la stagnation atteint un certain seuil on initialise la température à 6f par exemple et on remet la stagnation à 0, dans le but de sortir d'un minima local. Sachant que la température influe sur le critère de Métropolis. Et à chaque itération on décrémente la température.

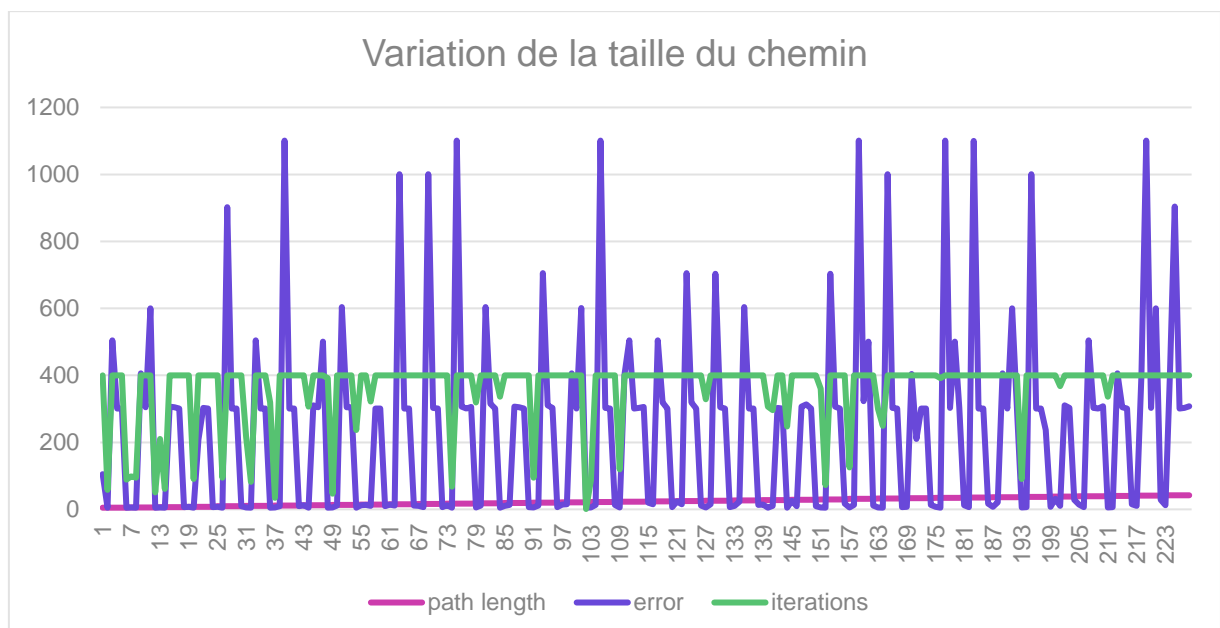
**Critère d'arrêt** : L'erreur est égale à l'erreur minimale.

### Valeur clé







**La taille du chemin** : idem que la recherche local naïve.

**La stagnation** : car cela définit le seuil de tolérance à un minima local. Trop grande et on reste trop longtemps sur une même solution, et trop bas on quitte trop vite le minima trouvé qui pourrait être le bon.

**La température** : influe sur la probabilité de bouger vers une meilleure nouvelle solution, et la probabilité de bouger vers une solution plus mauvaise est réduite en même temps que la température diminue.



# Algorithme génétique

PROBLEME	1	2	3	4	5	6
RESULTAT						

## Postulat

**Initialisation** : Génération d'un ensemble d'individu où un individu est un chemin de taille aléatoire.

**Evaluation** : Associe à chaque chemin un score.

**Selection** : Selectionne les meilleurs individus ~10% de la population.

**Croisement** : Deux parents aléatoire donne un enfant qui à une action sur deux de chaque.

**Mutation** : On sélectionne aléatoirement une des actions et en génère une nouvelle.

**Critère d'arrêt** : Le score du meilleure fils est égale à l'erreur minimale.

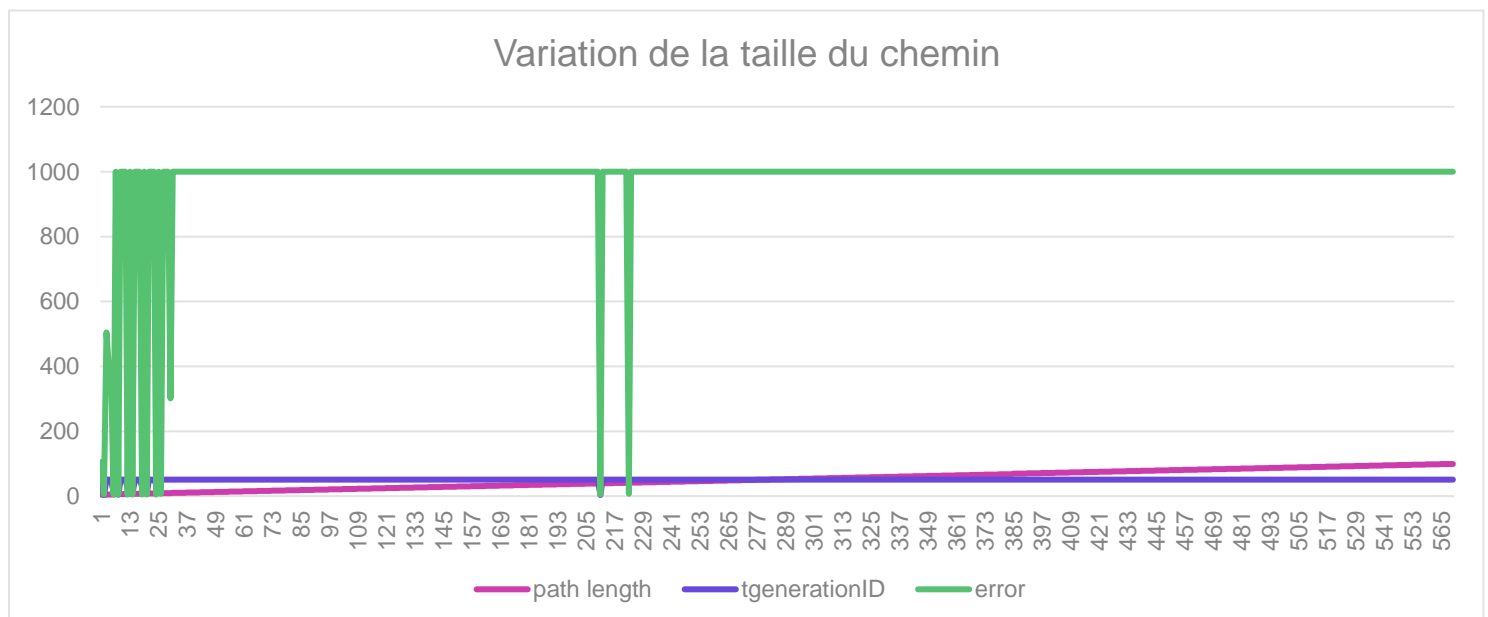
## Valeur clé

**La taille du chemin** : idem que pour les deux algorithmes précédents.







**La taille de la population** : une population trop faible tendra moins vite à trouver une solution, et une population trop forte sera trop lente à évaluer.

**Le critère de sélection** : si il est trop grand il prendra des individus avec un score pouvant être très faible et impacter la génération suivante.

**Le pourcentage de mutation** : il ne doit pas être trop grand sinon cela impacte trop les individus fils et les dégrade.



# Dijkstra

PROBLEME	1	2	3	4	5	6
RESULTAT						

## Postulat

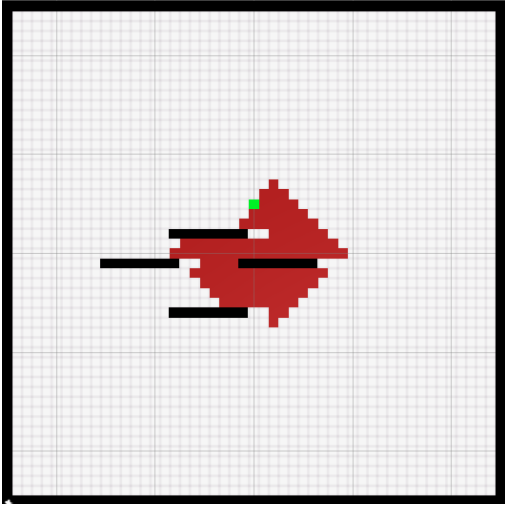
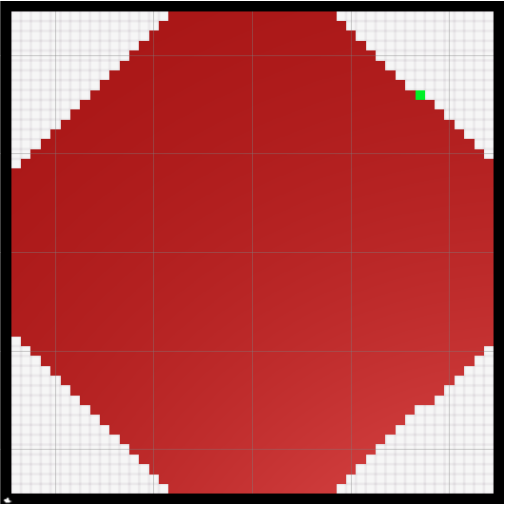
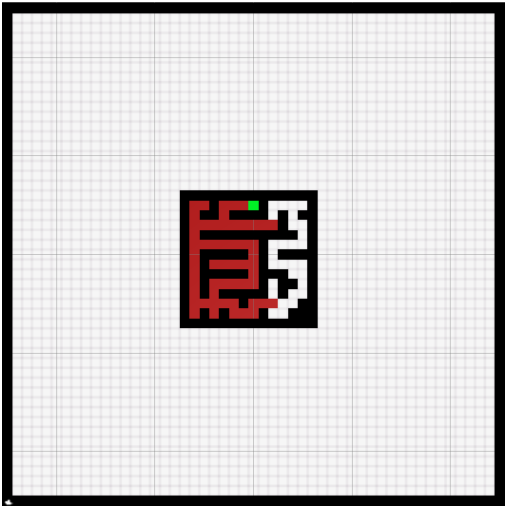
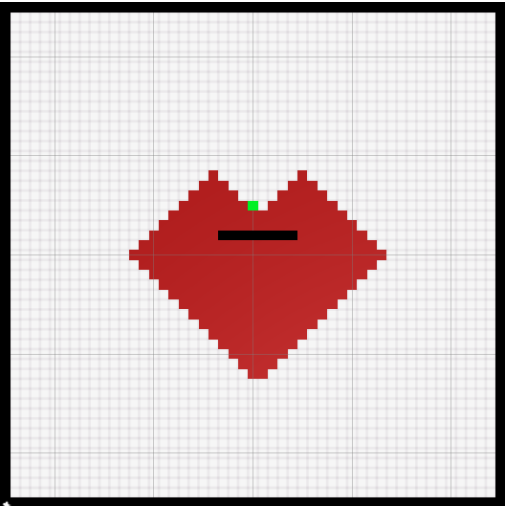
**Initialisation** : Etiqueter tous les nœuds du graphes avec un score infini. Etiqueter le nœud de départ avec un score nul. Ajouter tous les nœuds à la liste L.

**Boucle** : Choisir le nœud de score le plus faible dans L : Nmin.

Si Nmin == Nœud destination -> fin et remontée.







Pour chaque fils de Nmin : Mettre à jour le score selon la formule  $\text{score}(\text{Nmin}) + \text{Cout}(\text{Nmin} \rightarrow \text{Voisin})$  à condition que le résultat soit plus faible que  $\text{score}(\text{Voisin})$ .

Retirer Nmin de L



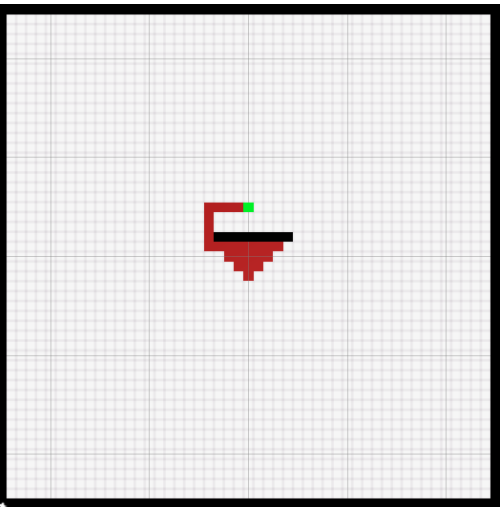
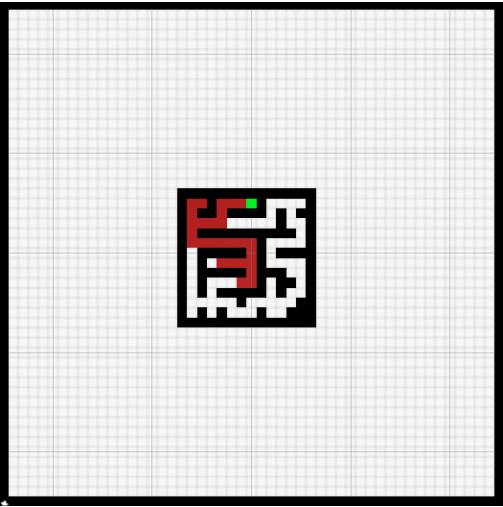
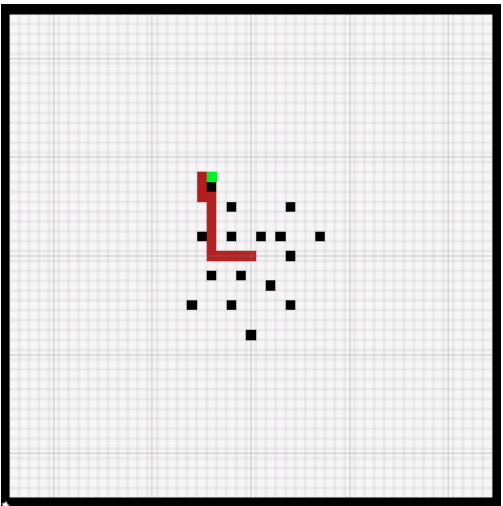
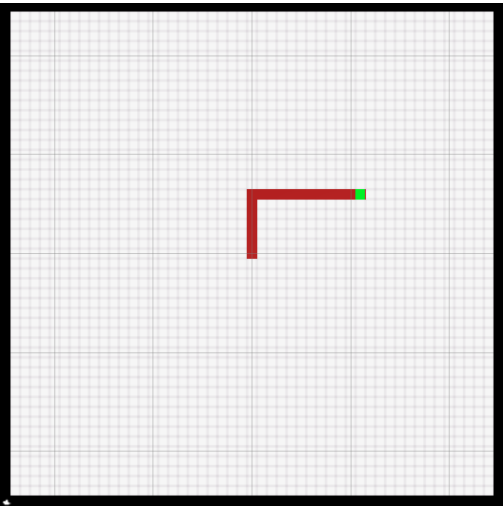
*Case rouge : case visité.*

# A Star

PROBLEME	1	2	3	4	5	6
RESULTAT						

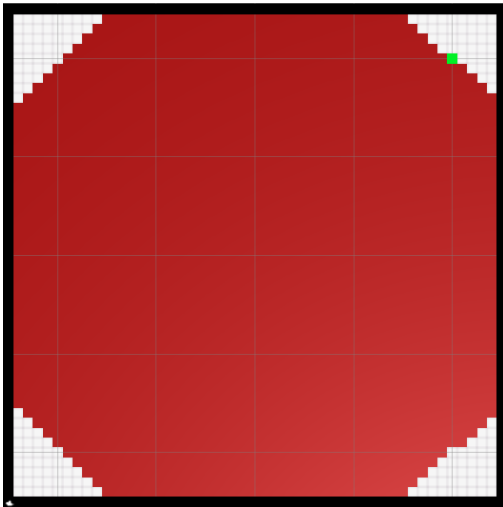
## Postulat

Identique que Dijkstra sauf qu'ici on ajoute la notion d'heuristique. On note chaque nœud avec celle-ci et on cherche le nœud nMin telque  $abs(score - \text{heuristique score})$  soit le plus faible. L'heuristique choisi ici sera la Distance de Manhattan.

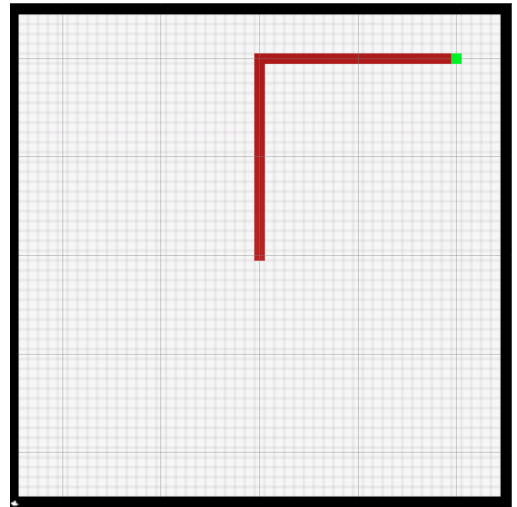


Case rouge : case visité.

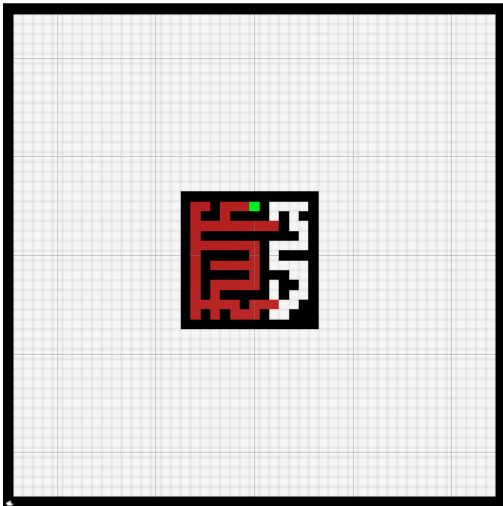
## Dijkstra vs A\*



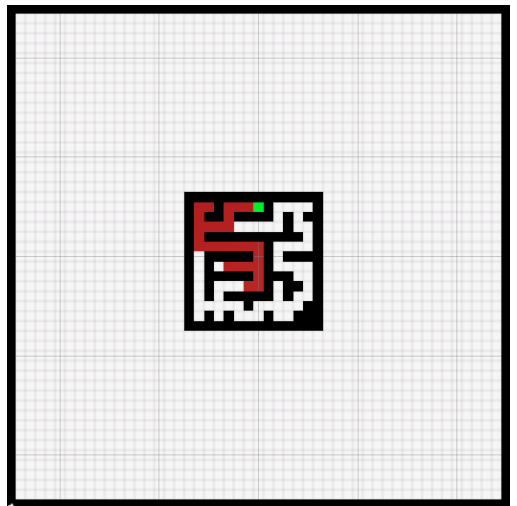
2229 itérations



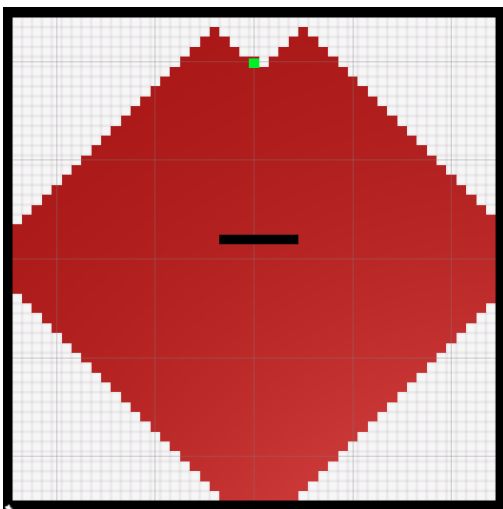
40 itérations



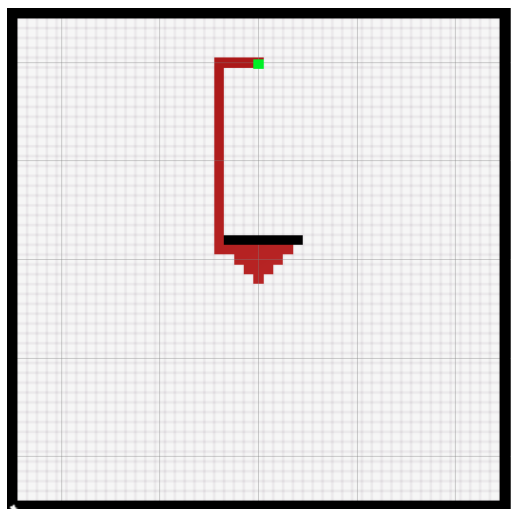
55 itérations



27 itérations



1482 itérations



39 itérations

*Case rouge : case visité.*