

Rapport d’avancement

Mai 2019

Projet tutoré JAVA

L’algorithme d’Ariane

Table des matières :

I/ Introduction

II/ Fonctionnalités

III/ Structure du programme

IV/ Présentation de l’algorithme déterministe

V/ Conclusions individuelles

I/ Introduction



Pour le Projet d’APL 2.1, l’objectif fut de concevoir un programme qui résout un labyrinthe à partir de nos connaissances en JAVA.

Dans le contexte, nous avons supposé que le personnage soit Thésée perdu dans le labyrinthe crétois et cherchant la sortie.

Notre programme devait avoir plusieurs fonctionnalités :

- Il se déroule en 3 parties : choix de grille, choix de l’algorithme, test de l’algorithme.
- Pour choisir la grille, l’utilisation d’un gestionnaire de fichier était obligatoire.

- Pour créer la grille, l'utilisateur peut partir d'une grille vide ou d'une grille remplie aléatoirement puis modifier chaque cellule indépendamment.
- L'utilisateur peut choisir entre un algorithme aléatoire et un algorithme déterministe.
- Ensuite, il peut choisir soit une visualisation automatique soit manuelle.

II/ Fonctionnalités

Pour répondre aux attentes en termes de fonctionnalités tout en apportant nos préférences, nous avons choisi d’organiser le programme de la manière suivante :

- Sur le menu, trois choix sont possibles :

Choisir une grille existante
Construire une grille
[Quitter]

Choisir une grille ouvre un sélecteur de fichier.

Construire une grille ouvre le constructeur de grille.

Quitter permet de clore le programme.

Ensuite, nos principaux choix sont au niveau du constructeur et du menu avant de lancer l’algorithme puisque le constructeur propose des boutons vider et random pour générer autant de grilles temporaires que l’on veut avant de valider au lieu de choisir random ou vide au début de la construction. Aussi, nous avons choisi de placer Thésée et la sortie lors de l’appui sur random. Cela n’empêche pas de modifier leur position ensuite. D’ailleurs, le choix des actions permettant de créer une grille est le suivant :

Un clic change l’état de la cellule :

Etat 0 = chemin (blanc)

Etat 1 = chemin (noir)

Etat 2 = chemin (rouge)

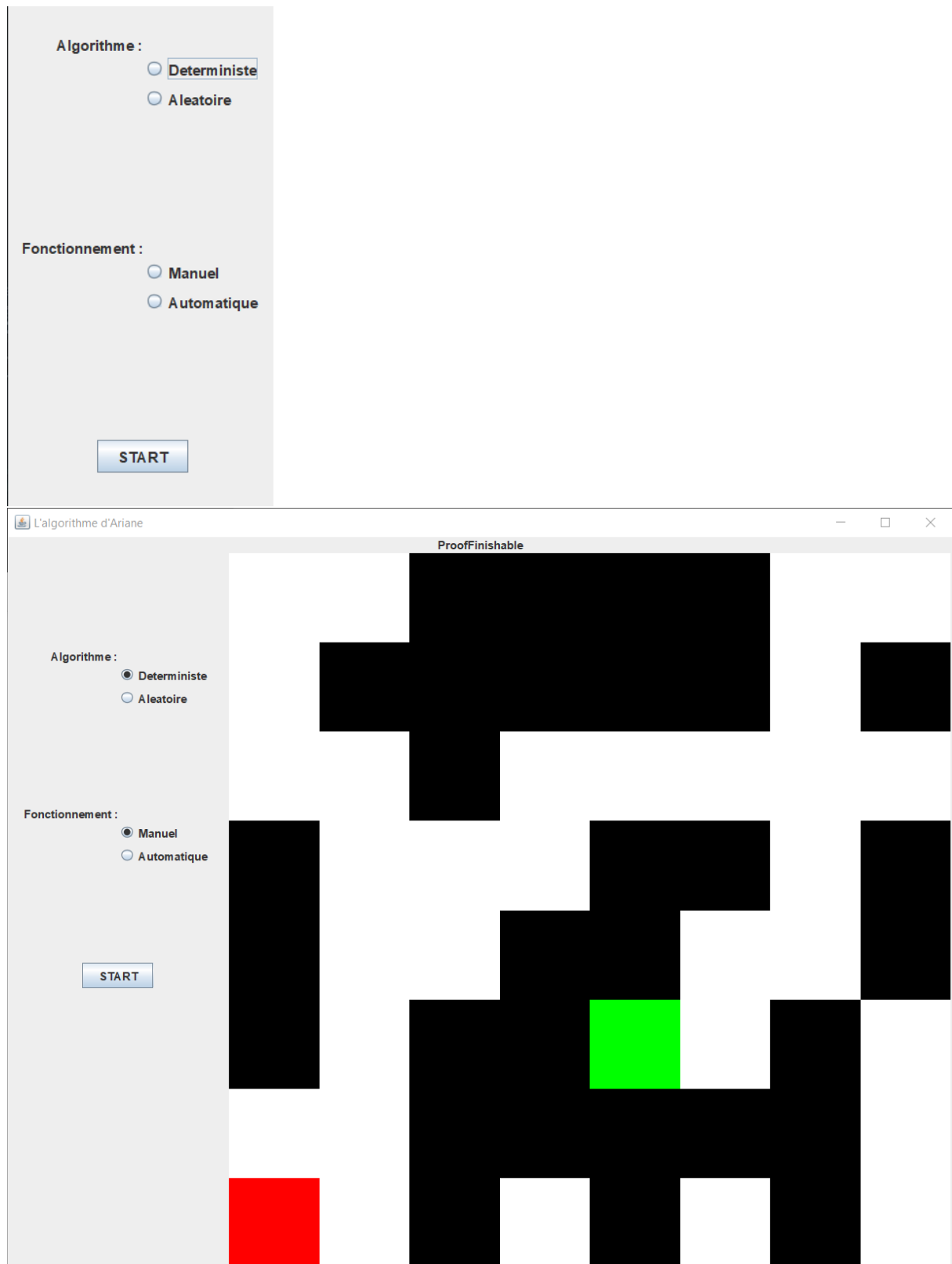
Etat 3 = chemin (vert)

Voici un échantillon de la représentation des différents états possible :



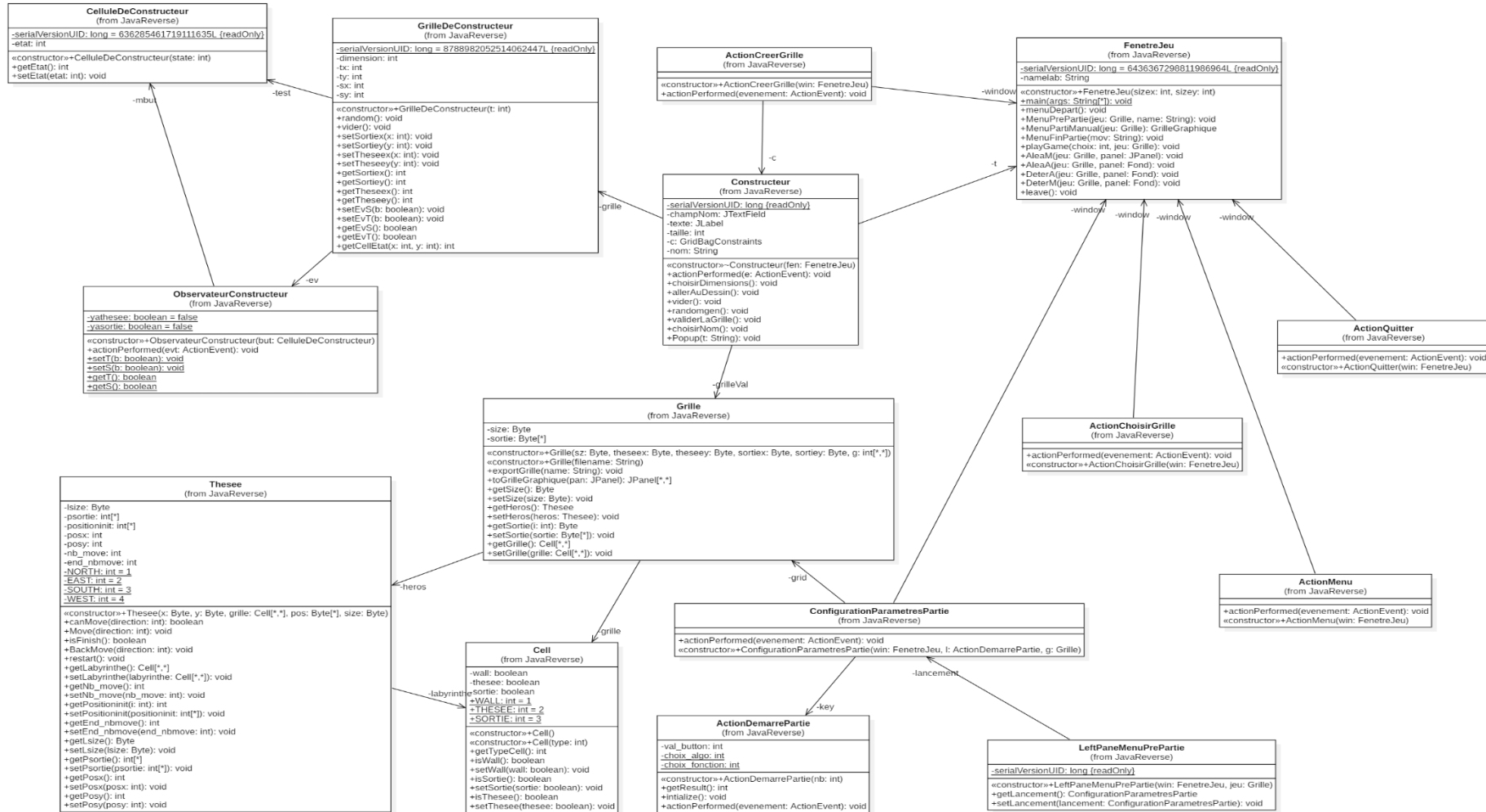
Pour ce qui est de la sauvegarde de la grille créée, un simple champ de texte est à remplir. Celui-ci filtre les caractères indésirables et sauvegarde automatiquement dans le dossier prévu à cet effet.

Ensuite, pour le choix de l’algorithme et de la vue, nous avons choisi des boutons radio, qui nous semblaient plus adaptés que des interfaces avec des boutons qui s’enchainent.



III/ Structure du programme

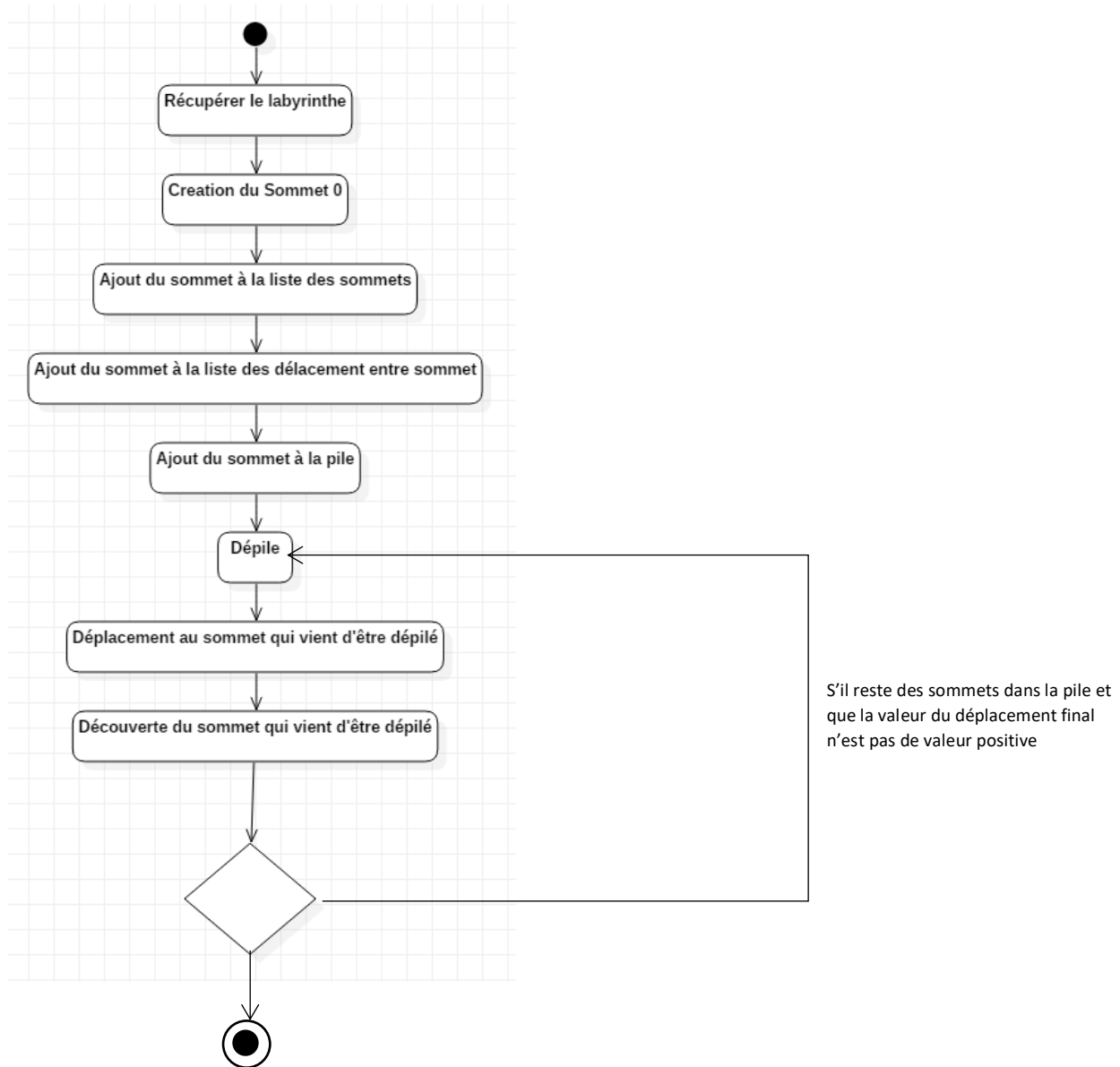
Notre programme se déroule de la manière suivante : (Partie Configuration)

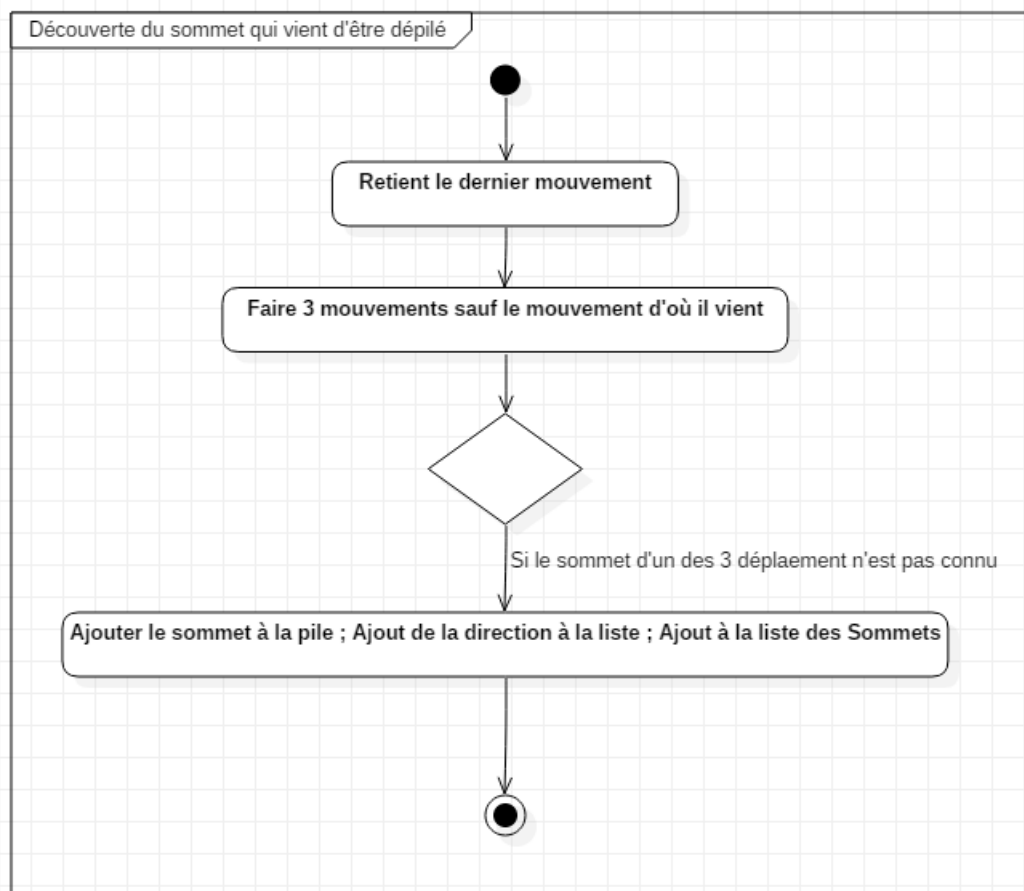
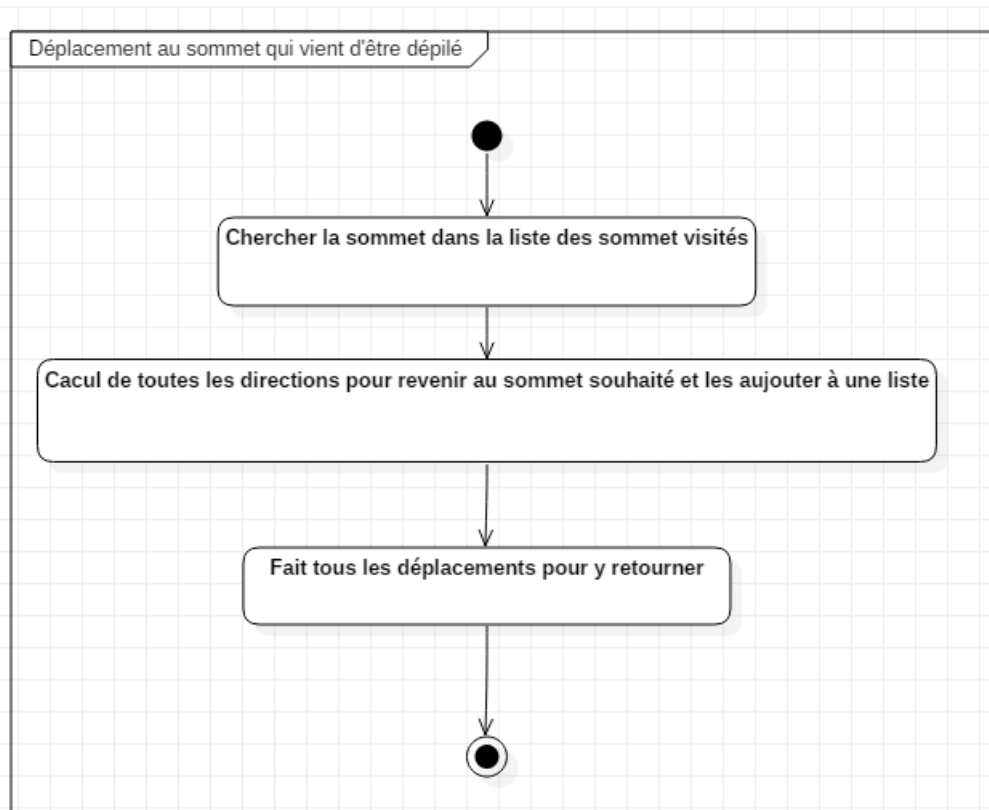




IV/ Présentation de l’algorithme déterministe

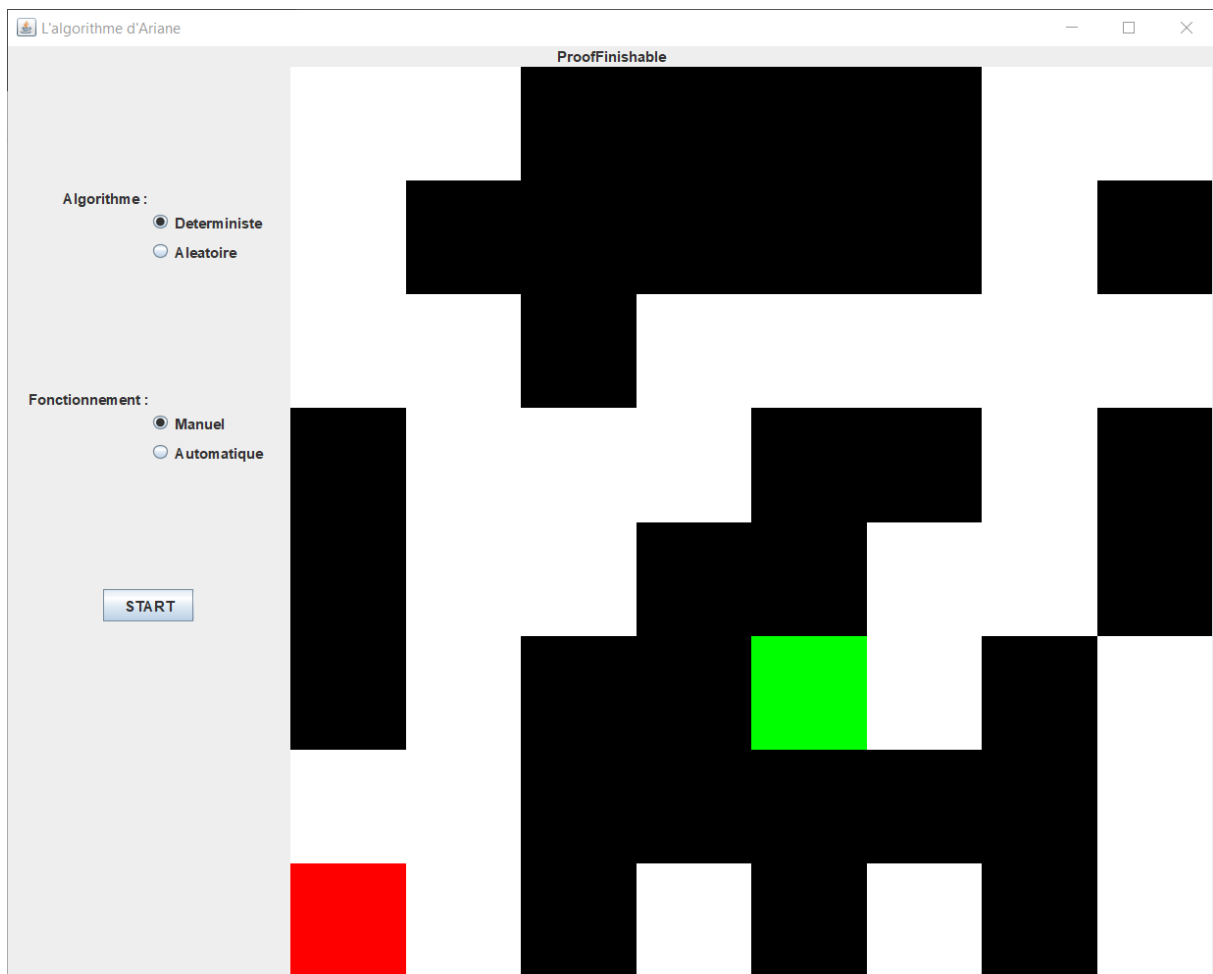
Pour l’algorithme déterministe, il est explicable avec les diagrammes suivants.





Pour prouver l’efficacité de l’algorithme, voici quelques cas :

Ici, une petite grille.



Le resultat est probant, l’algorithme a reussi.

Nombre de coups en Deterministe : 79

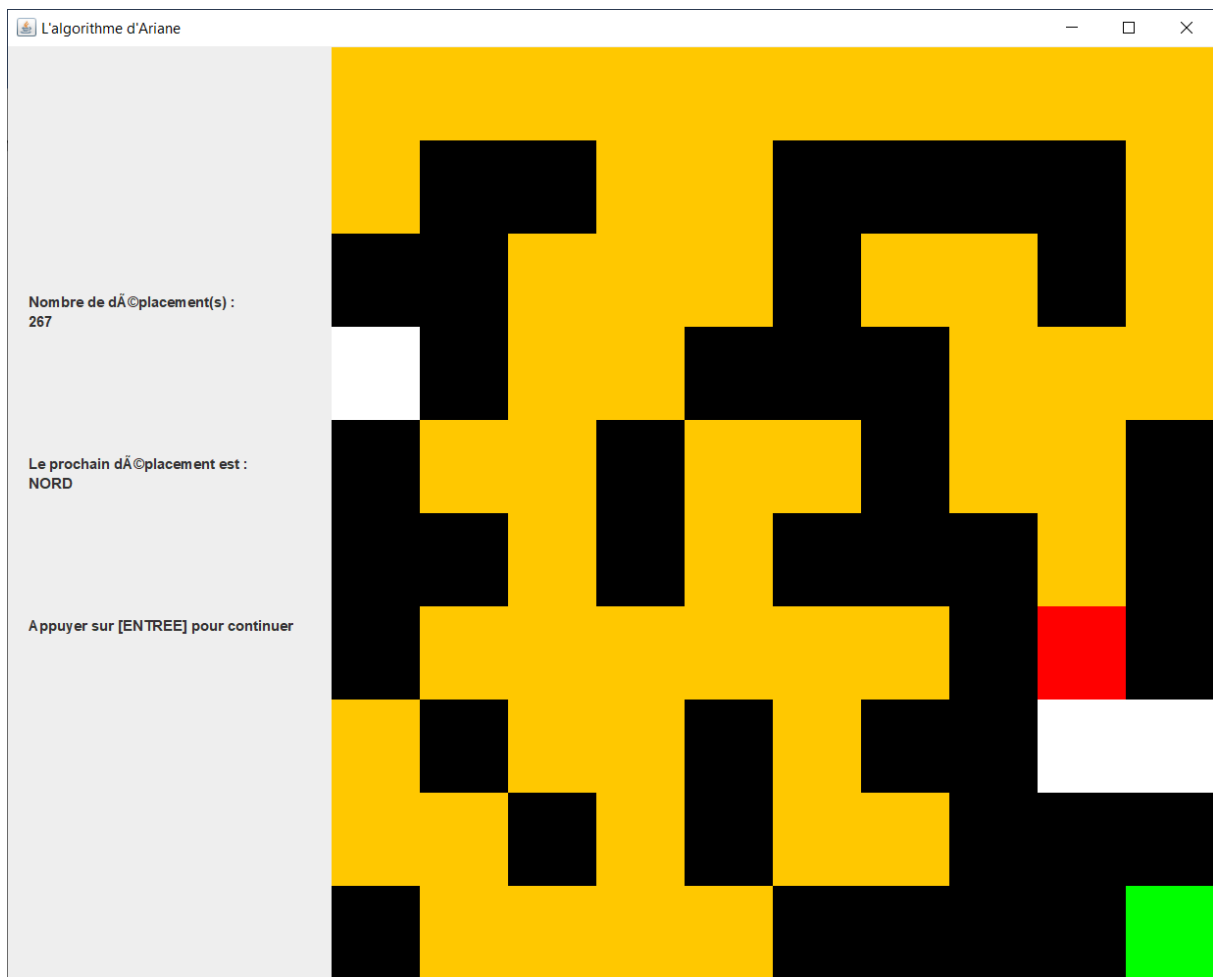
Ensuite, nous allons tester une grille plus conséquente de 120x120, toujours terminable.



Une fois de plus, l’algorithme a trouvé la sortie.

Maintenant, voyons un cas où la sortie n’est pas joignable par Thésée : même chose en bloquant la sortie. Si Thésée a parcouru toutes les cases, c’est-à-dire qu’il revient uniquement sur des cases explorées, alors on admettra que le labyrinthe est insolvable.

Ici, le parcours avance, et la sortie n’est pas joignable :



Le résultat est sans appel : pas de solution :

Nombre de coups en Deterministe : Impossible

V/ Conclusions personnelles

Maxime :

Ce projet a été pour moi une nouvelle expérience, le code sous forme d'objet est une chose que je n'avais pas utilisé encore pour un projet. Ce projet je l'ai réalisé en duo avec Théo qui m'a été d'une grande aide pour la construction de mes classes, notre duo a bien fonctionné dans l'ensemble. Nos tâches ont été réparties en classe ce qui nous a permis d'avoir une intégration plus que simple. La partie qui m'a le plus intéressée était l'algorithme déterministe qui m'a permis d'utiliser mes connaissances des graphes des mathématiques appris cette année. Au final, ce projet m'a permis de mieux comprendre l'utilisation du java dans un projet.

Théo :

De mon côté ce projet a été l'occasion de découvrir et de mieux assimiler les points de la programmation orientée objet qui ne m'étaient pas familiers et de pouvoir me « mettre à niveau » puisque le concept de classe et de passage d'attribut me faisait parfois défaut. De plus, le travail en binôme m'a bien fait comprendre l'utilité des classes pour la répartition du travail et la collaboration. De plus, la création de commentaires compatibles javadoc m'a fait comprendre la puissance de cet outil pour rendre ses propres classes utilisables. En conclusion, je suis assez satisfait de ce projet même si j'aimerais l'améliorer en apportant un peu plus d'interaction entre l'utilisateur et le logiciel, par exemple avec un mode où l'utilisateur doit résoudre une grille plus rapidement que l'algorithme déterministe.