Projet Tutoré 2.1
PT21_APL2018

L'ALGORITHME D'ARIANE

Nicolas FAFIN & **Antoine MAN**

Sommaire

Introduction

Le projet en java de cette année porte sur l'étude d'un algorithme de guidage, visant à conduire un objet mobile jusqu'à son but à travers un parcours d'obstacle. Et pour illustrer ce projet, il était énoncé de rendre hommage à la mythologie grecque, en supposant qu'il s'agissait de Thésée, perdu dans le labyrinthe crétois, à la recherche de la sortie. Bien évidemment sans l'aide d'Ariane dans notre cas.

Petit point mythologie :

Minos, roi de Crète, pria le dieu Poséidon et lui demanda de lui envoyer un taureau qu'il sacrifierait en sa gloire. Aussitôt, un magnifique taureau blanc surgit du plus profonds des océans. Ébahi par la beauté de ce taureau, il décida de le garder et de sacrifier à la place un taureau quelconque de son troupeau, pensant que Poséidon ne remarquera rien. Malheur aux personnes qui essayent de tromper les dieux. En effet, Poséidon découvrit la supercherie et se mit dans une colère noir. Il lança alors une malédiction à la femme de Minos qui l'obligea à tomber folle amoureuse du taureau blanc. De leur union naquit le Minotaure, un affreux monstre mi-homme, mi-taureau. Minos décida alors d'enfermer cet horrible créature dans un immense labyrinthe. Thésée se porte alors volontaire pour aller tuer le Minotaure car chaque année, des personnes de son peuple sont envoyées dans le labyrinthe en sacrifice au Minotaure. Alors que Thésée doit entrer dans le labyrinthe, Ariane (fille de Minos) lui donne une bobine de fil pour qu'il puisse retrouver son chemin. Thésée s'engouffre dans le labyrinthe et finit par tomber sur le Minotaure. Avec son poignard, Thésée parvient à la tuer et parvient également à sortir du labyrinthe grâce au fil d'Ariane.

Le projet est structuré en trois parties :

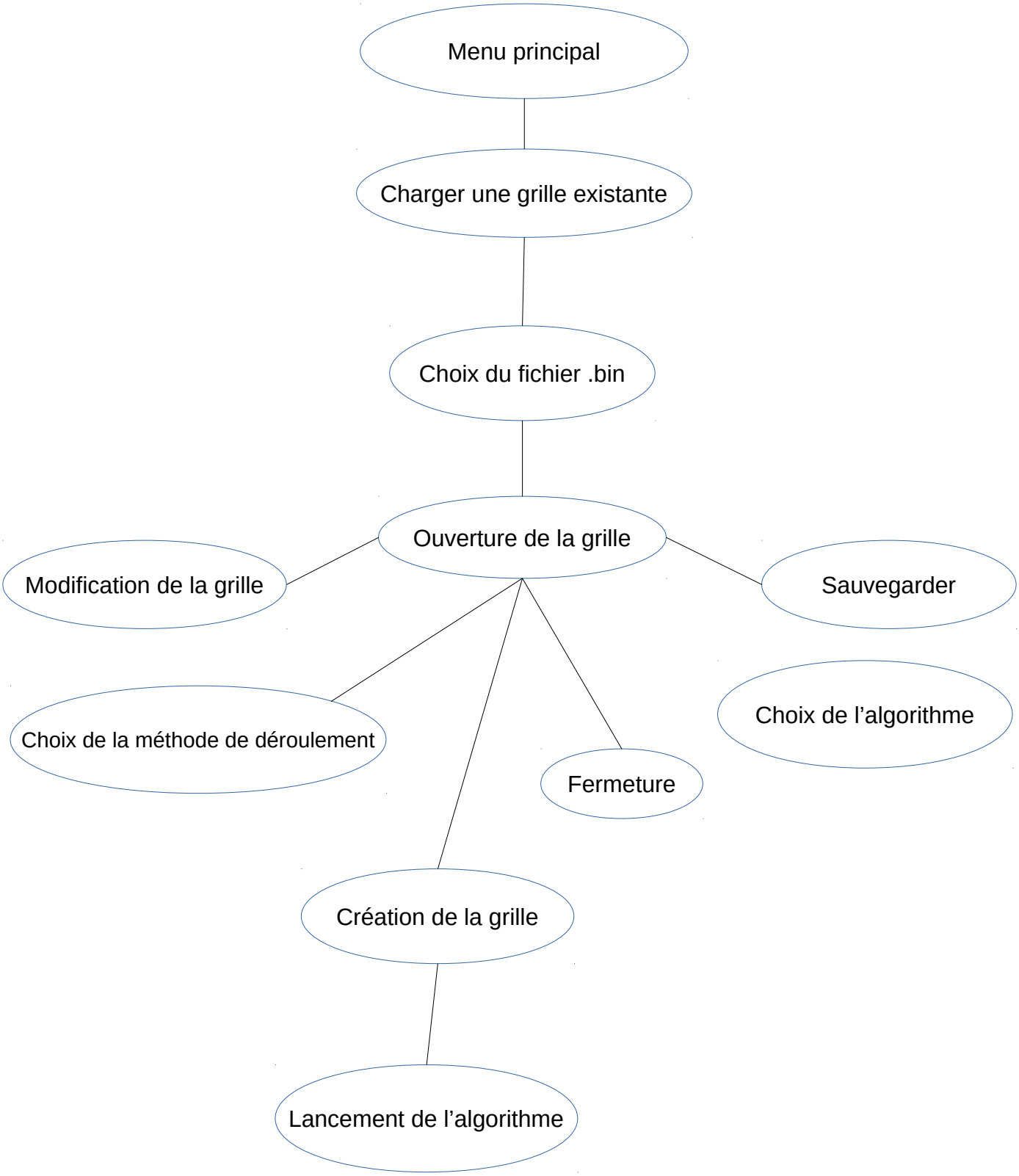
- L'utilisateur a la possibilité de charger un labyrinthe préalablement sauvegardé, ou de construire, lui-même un tout nouveau labyrinthe.

- L'utilisateur a ensuite le choix entre deux algorithmes, l'un totalement aléatoire et l'autre déterministe.

- Une fois le labyrinthe créé et l'algorithme choisi, il fera le choix entre une simulation automatique ou une simulation manuelle.

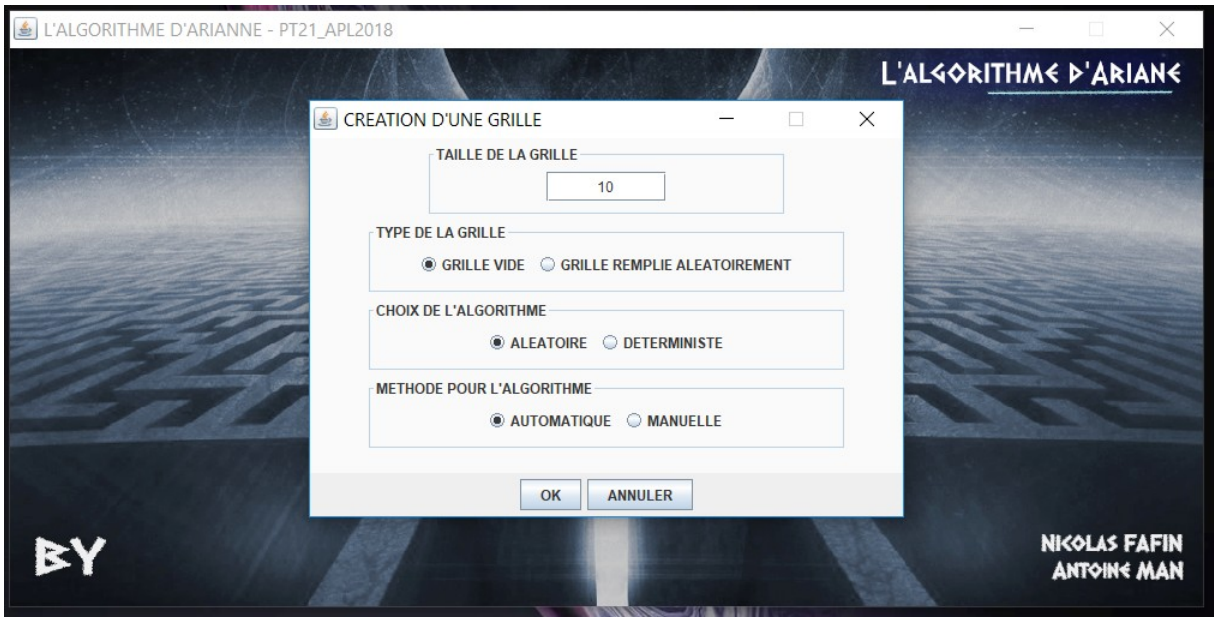
Fonctionnalités du programme

Dans le cas de l'ouverture d'une grille préalablement sauvegardée :



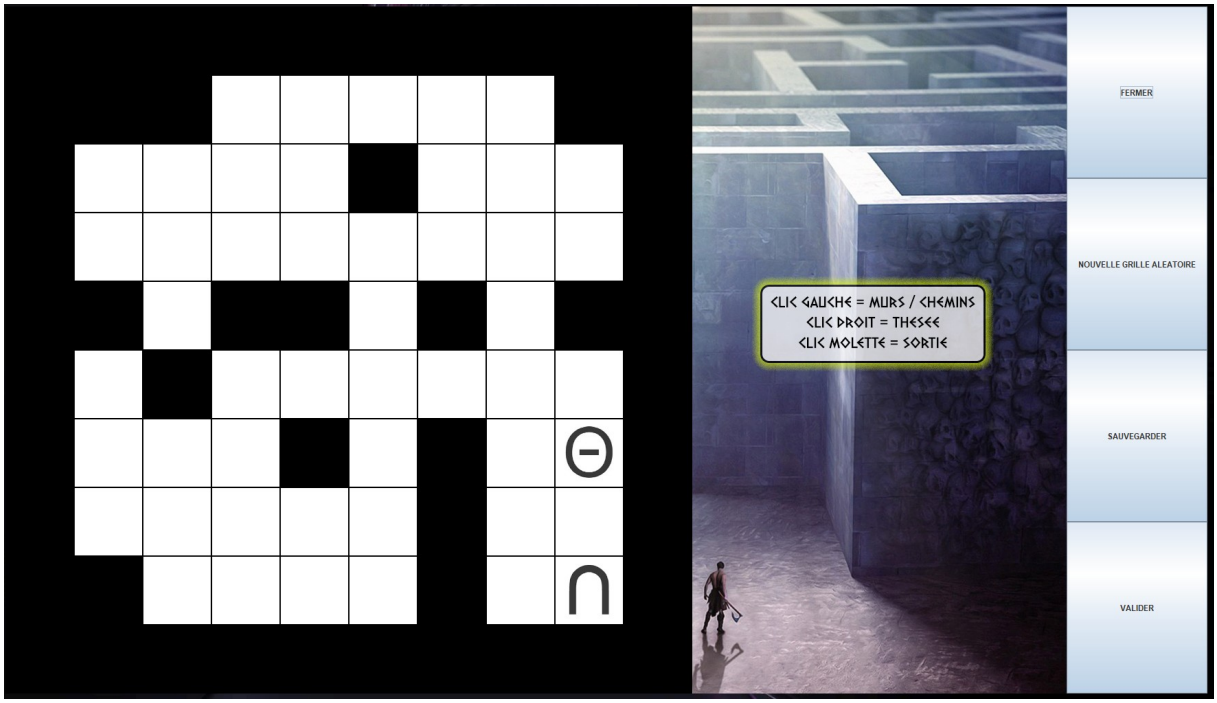
Dans le cas de la création d'une nouvelle grille :

Au lancement de notre programme, notre menu principal s'affiche et nous laisse le choix entre la création d'une nouvelle grille ou le chargement d'une grille existante dans notre dossier courant.

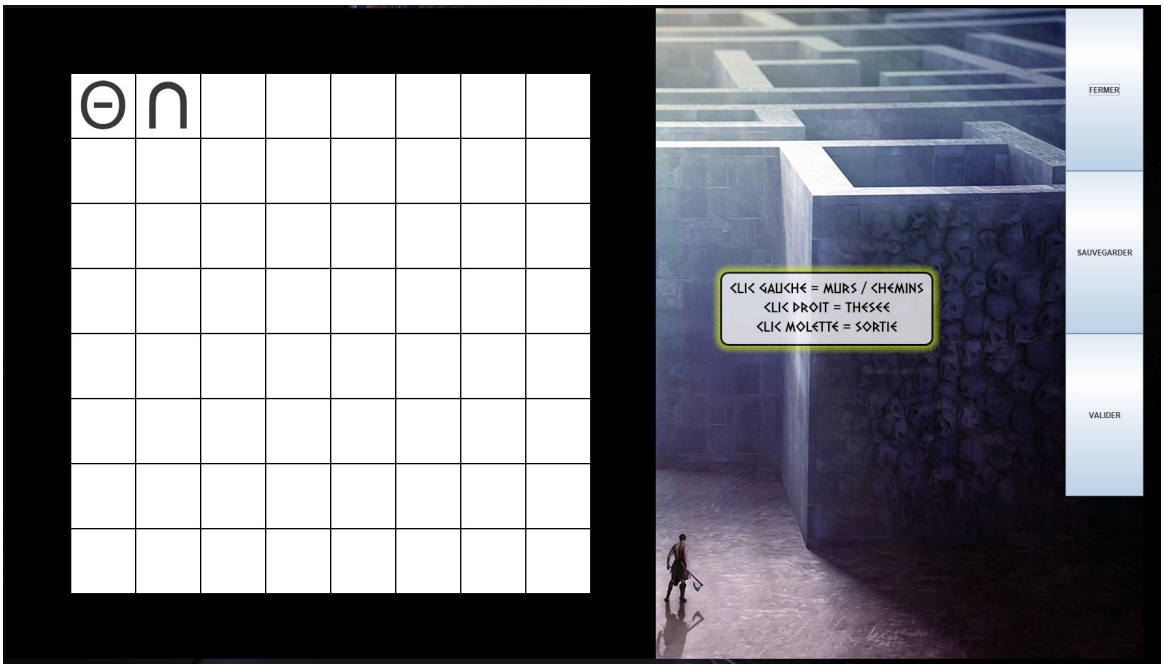


Si l'utilisateur souhaite créer sa grille, ce menu s'affiche.

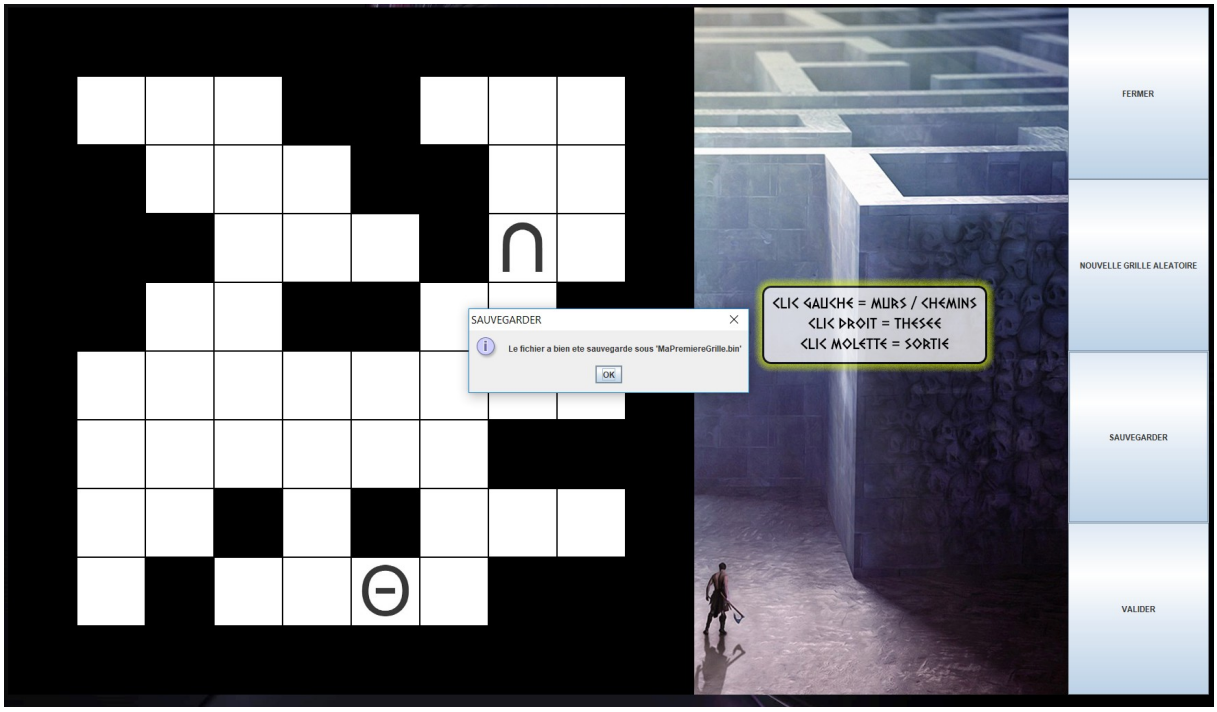
Il peut alors choisir la taille, l'algorithme utilisé et la méthode pour la simulation de cet algorithme.



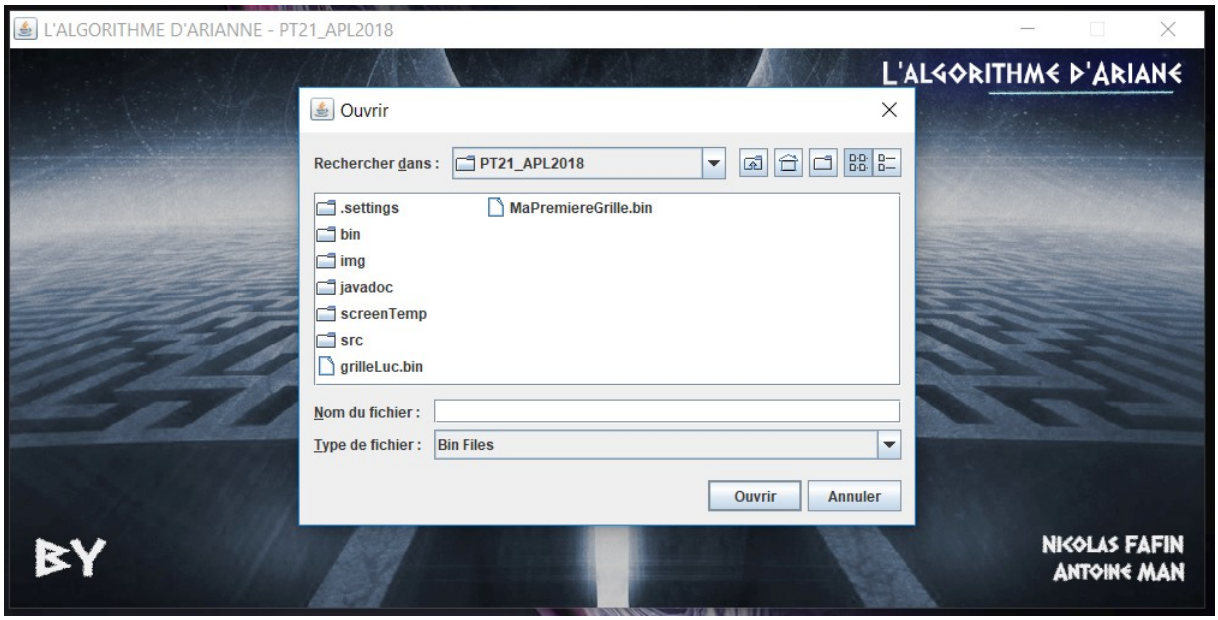
Si le choix de l'utilisateur se porte vers 'Grille aléatoirement remplie', alors une grille est générée aléatoirement. Il pourra par la suite, la modifier, la sauvegarder et la valider ou générer une nouvelle grille.



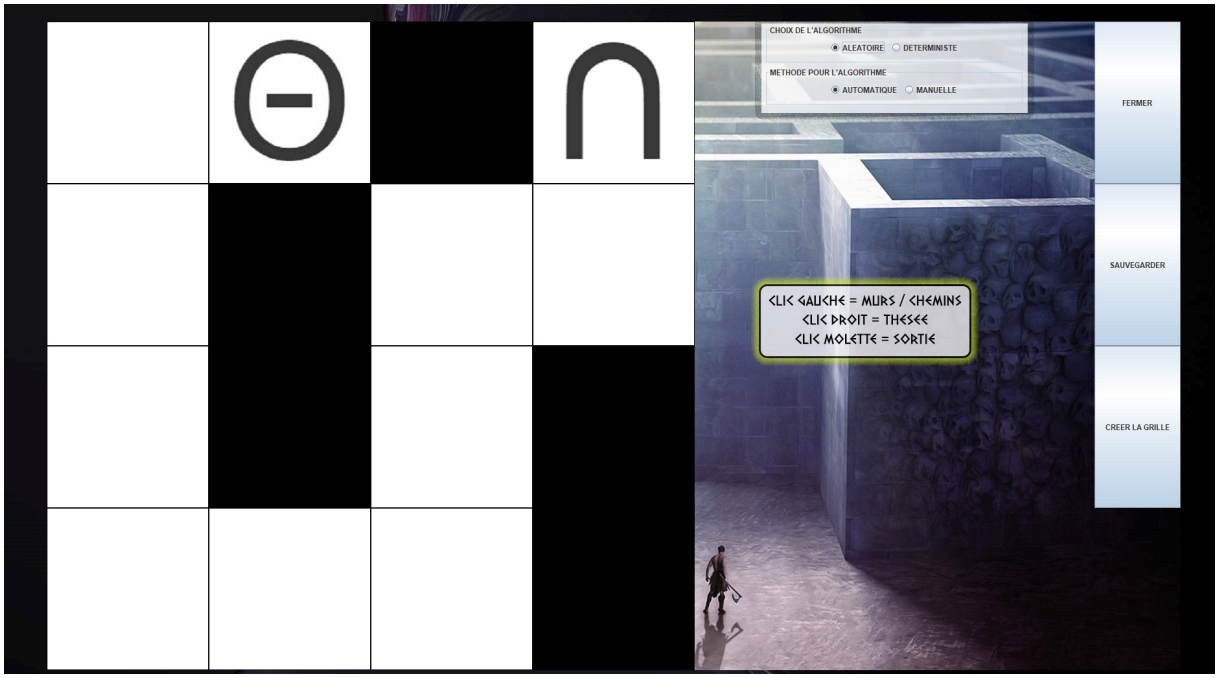
Dans le cas d'une grille vide, l'utilisateur peut créer de A à Z son propre labyrinthe. Lorsque le labyrinthe lui convient, il peut alors décider de la sauvegarder ou de la valider.



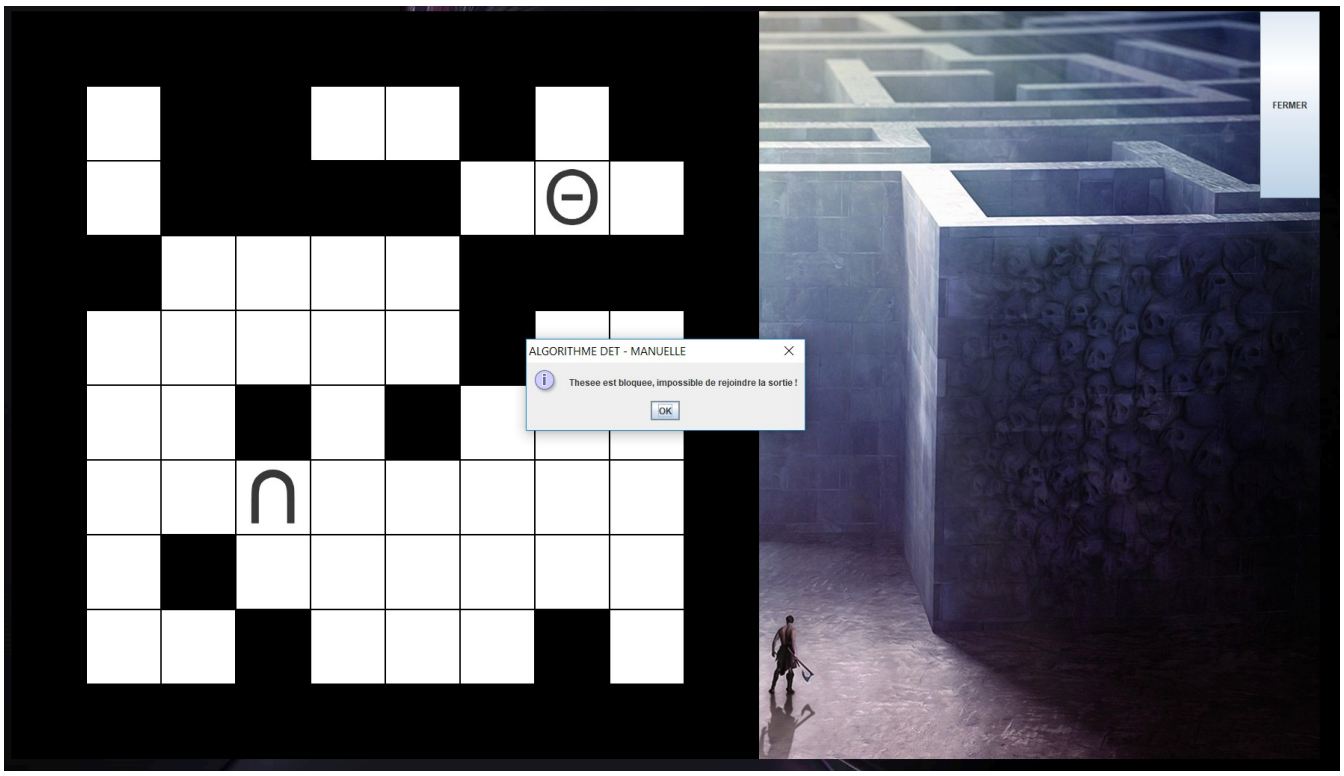
Lorsque l’on utilise la fonctionnalité sauvegarder le programme nous propose de choisir le nom de notre fichier puis le sauvegarde dans le dossier courant au format ‘.bin’.



L'utilisateur peut ouvrir une grille qu'il a créé et sauvegardé auparavant.

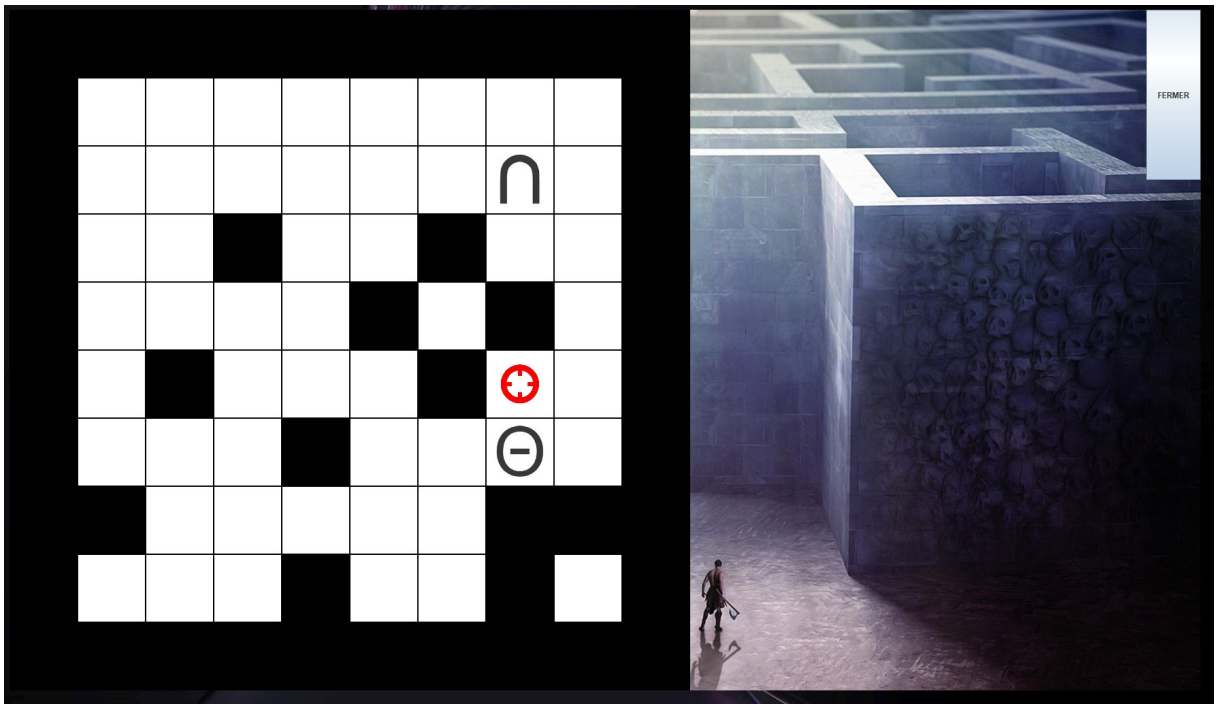


L'utilisateur peut modifier et choisir l'algorithme avant sa création définitive. Il peut également sauvegarder cette grille.



Lorsque l'utilisateur fournit une grille impossible à résoudre, alors nos algorithmes détectent qu'il n'y a aucunes solutions et affiche un message d'erreur.

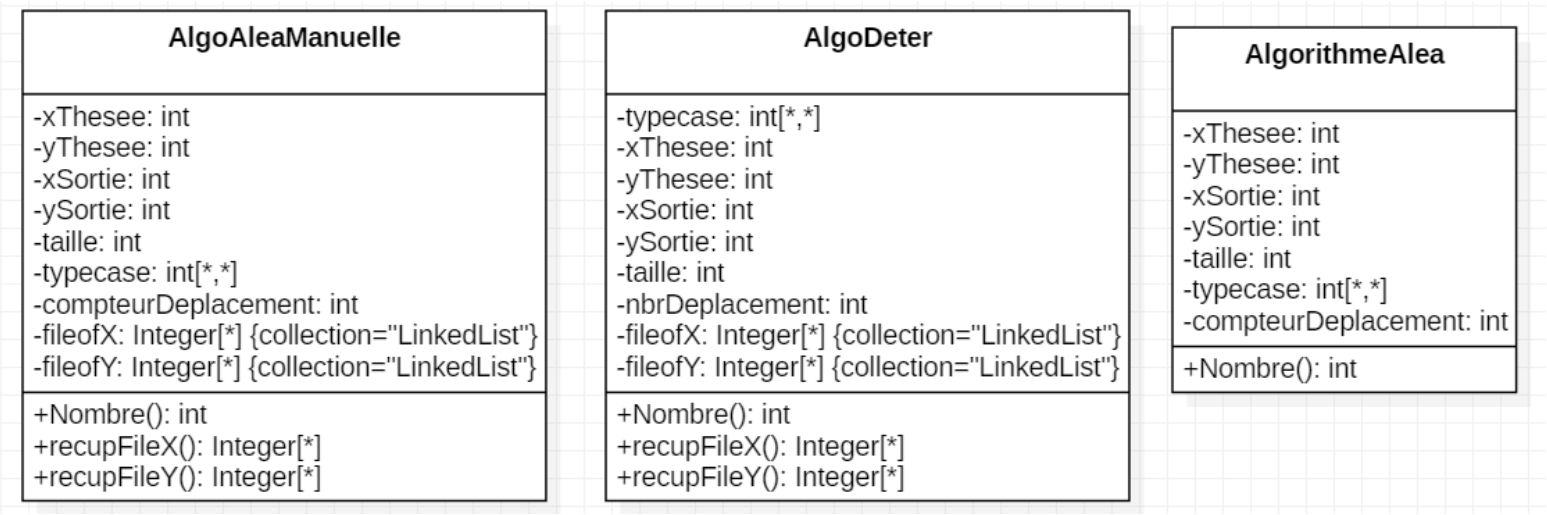
Dans le cas contraire, en mode automatique, l'algorithme nous renvoie une moyenne du nombre d'étapes après une simulation de 100 essais.



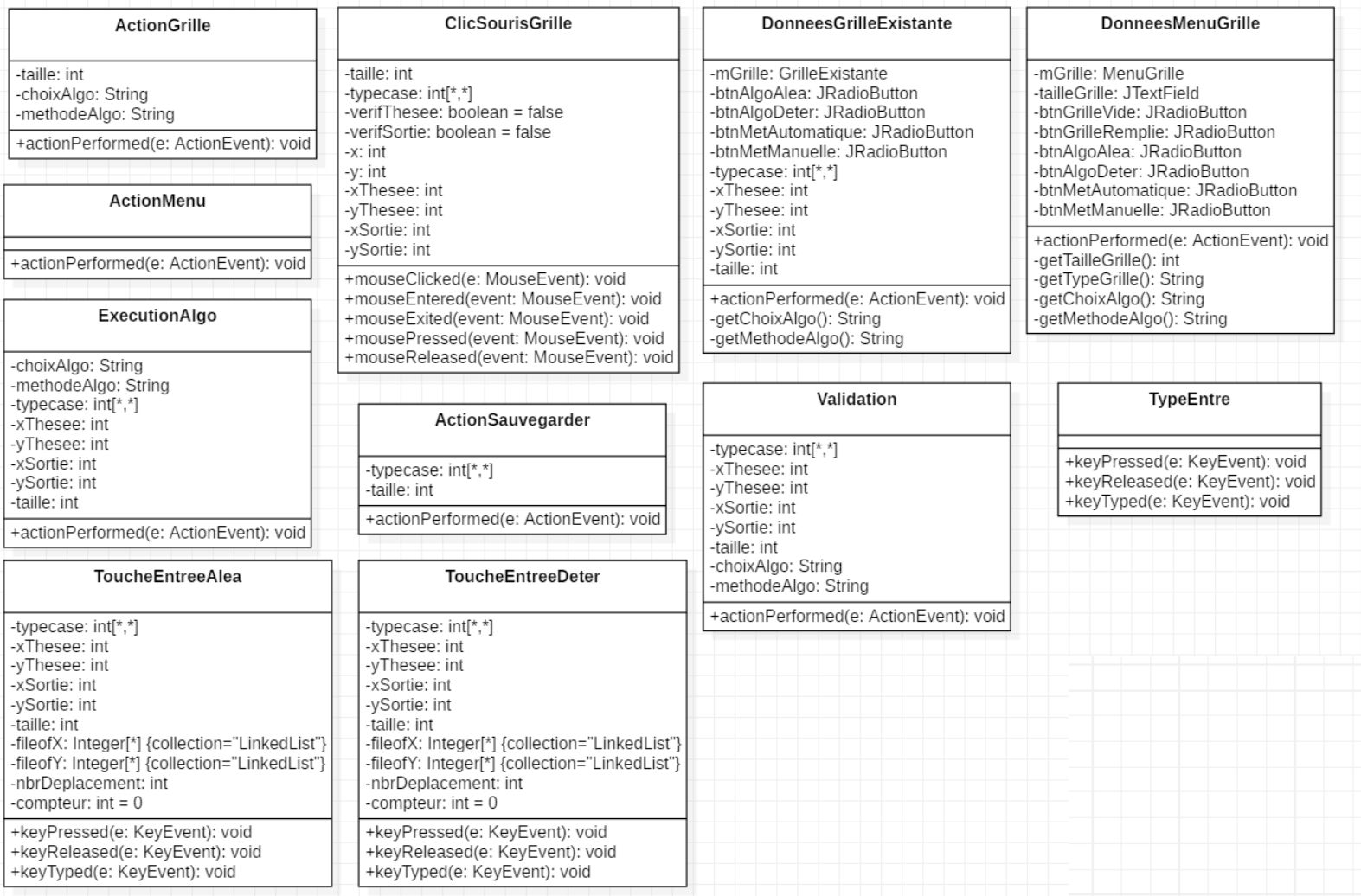
Concernant le mode manuel, l’algorithme nous détaille le chemin emprunté par Thésée et nous désigne la prochaine étape que Thésée effectuera par la cible en rouge.

Structure du programme

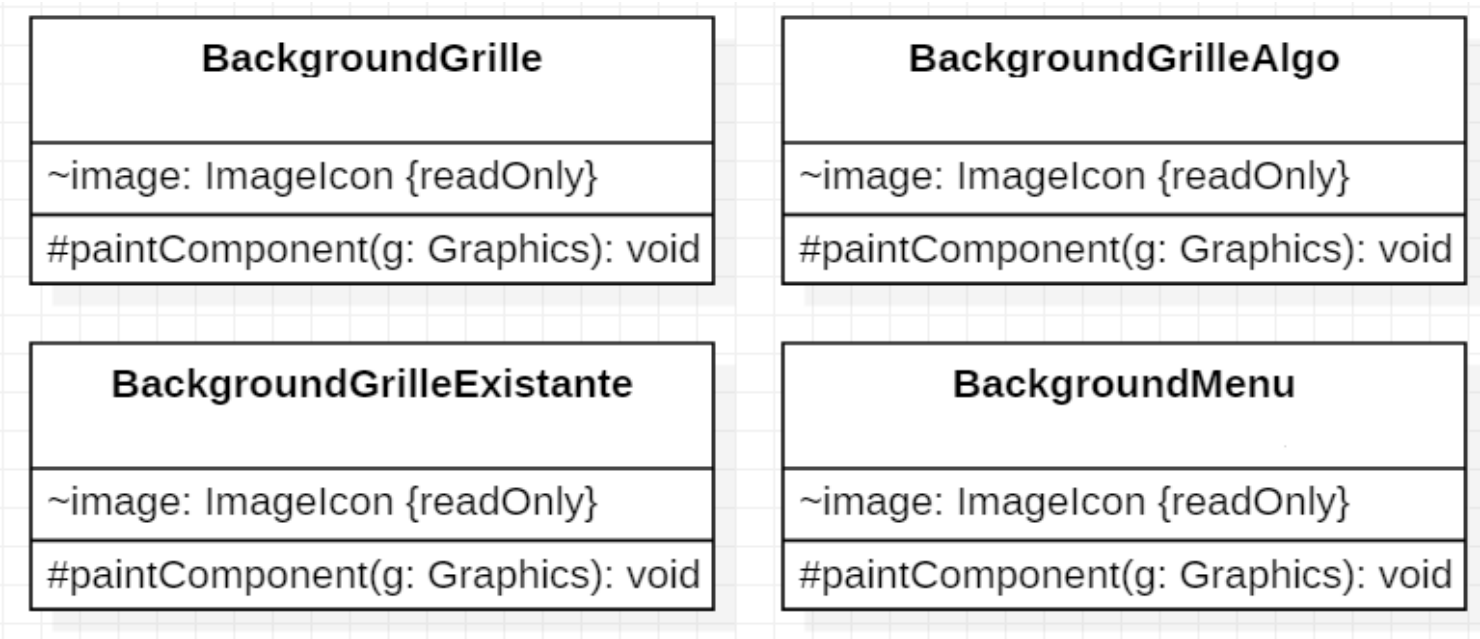
Classes Algorithmes



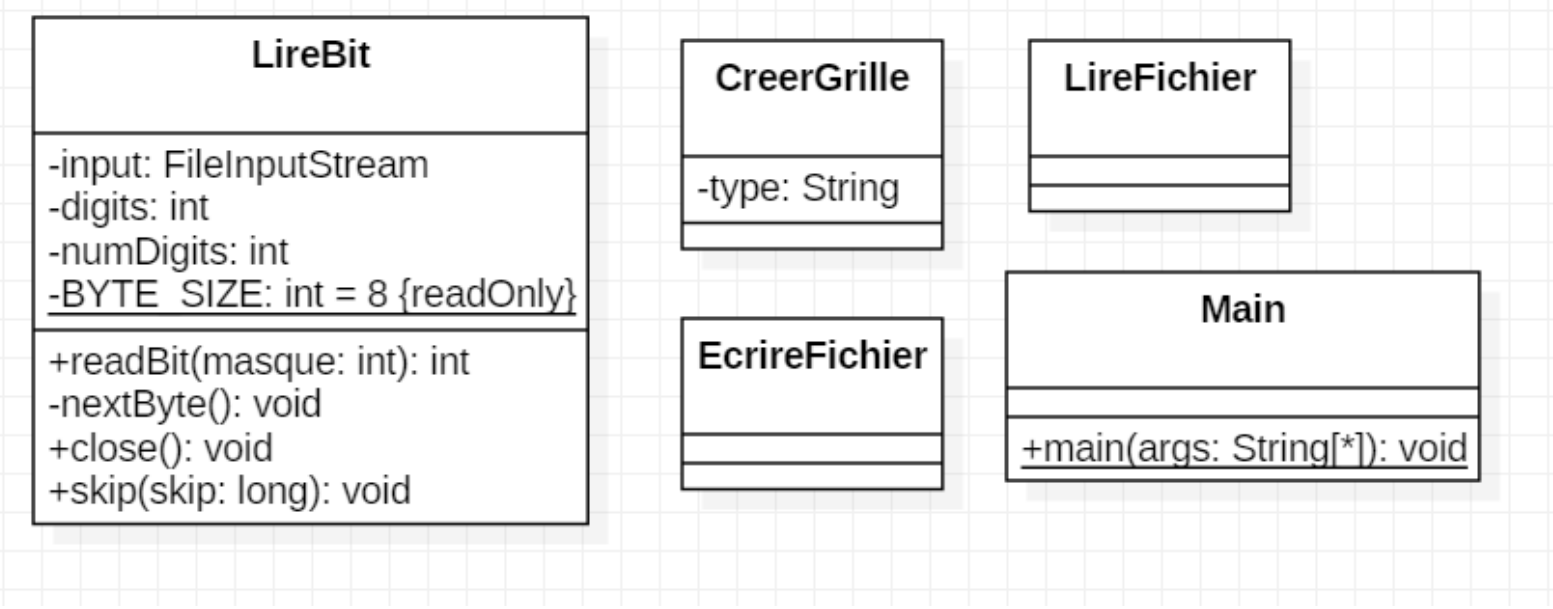
Classes Contrôler



Classes Background



Classes Modèles



Classes Vues

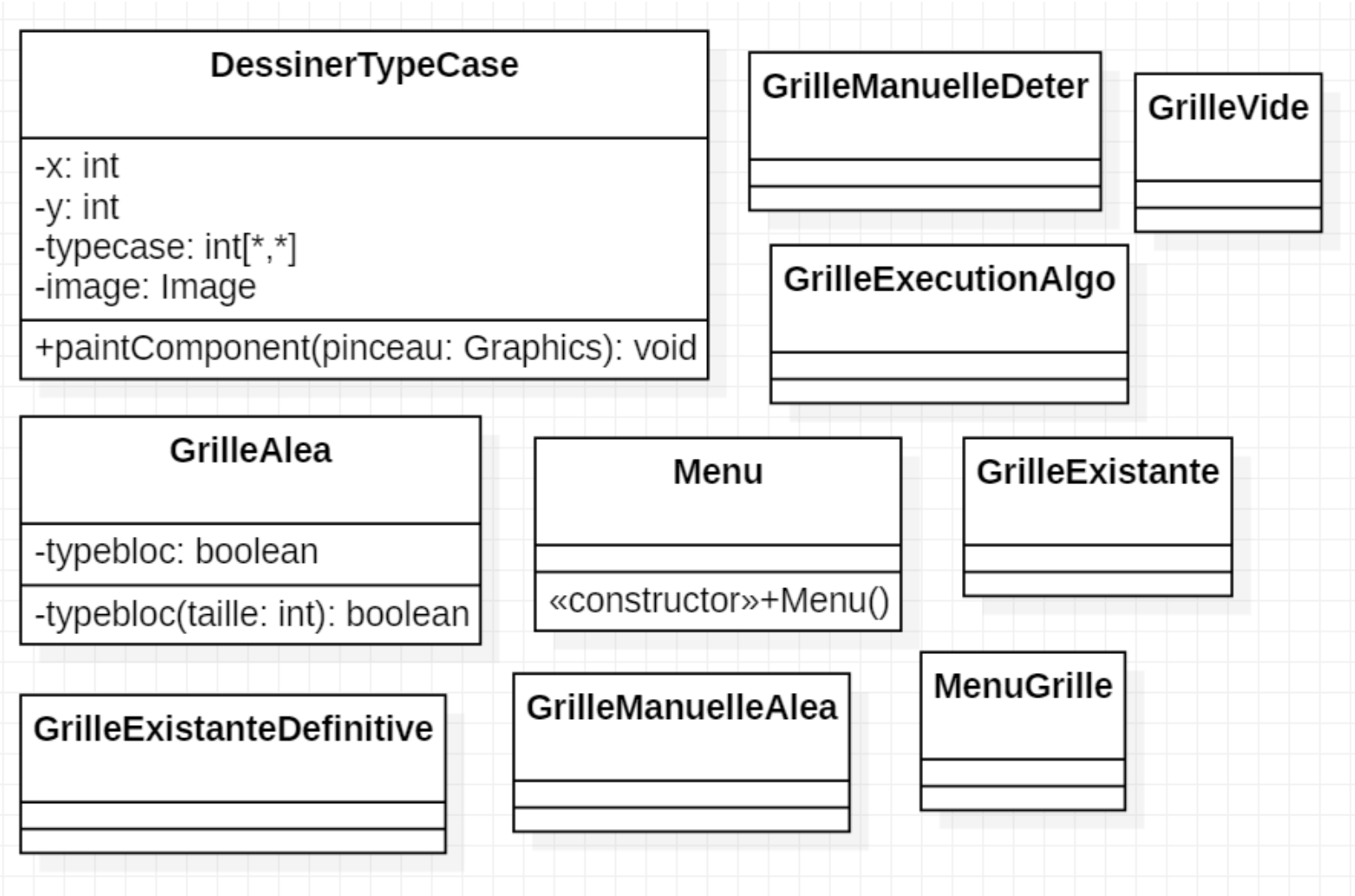
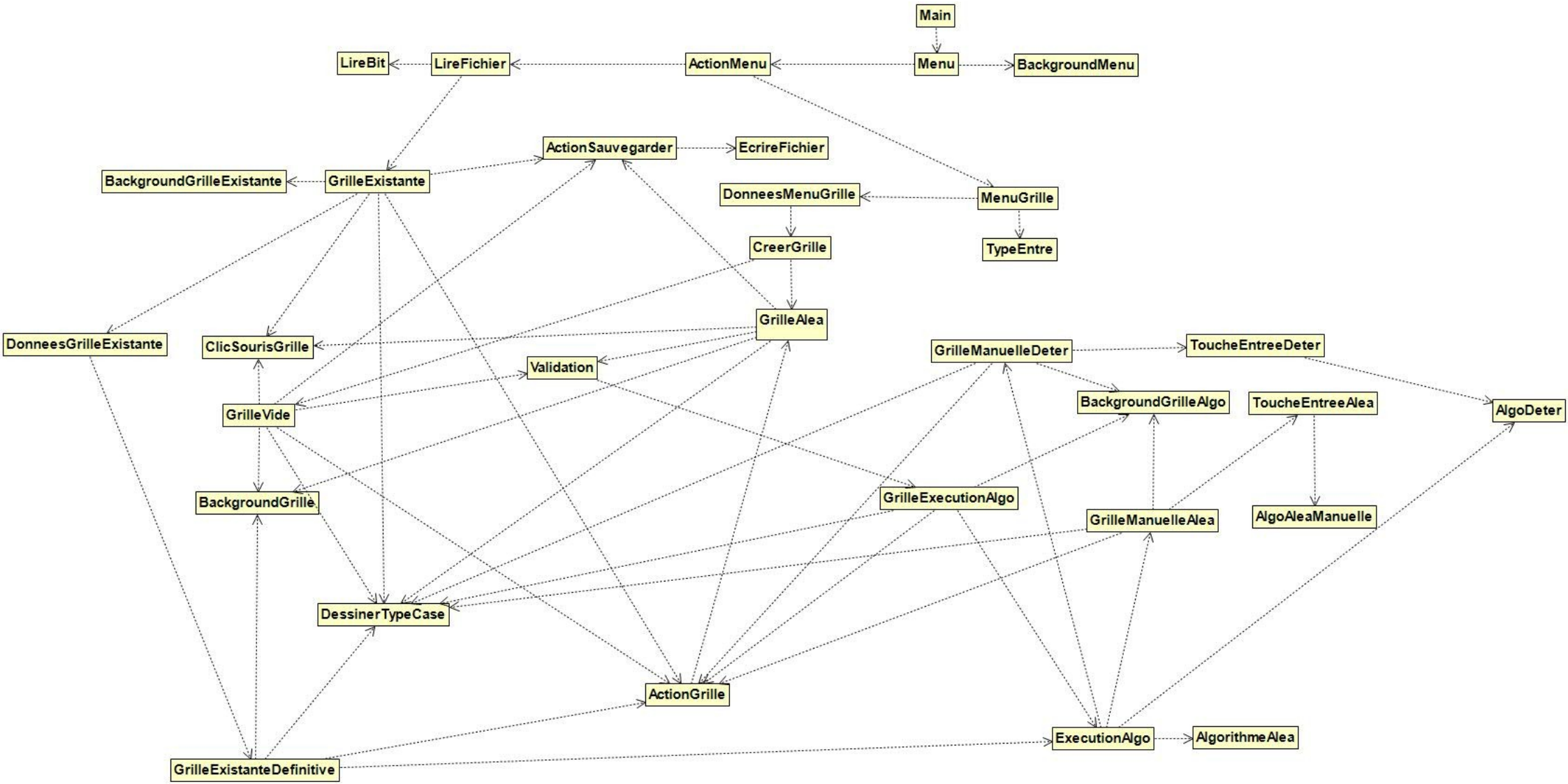
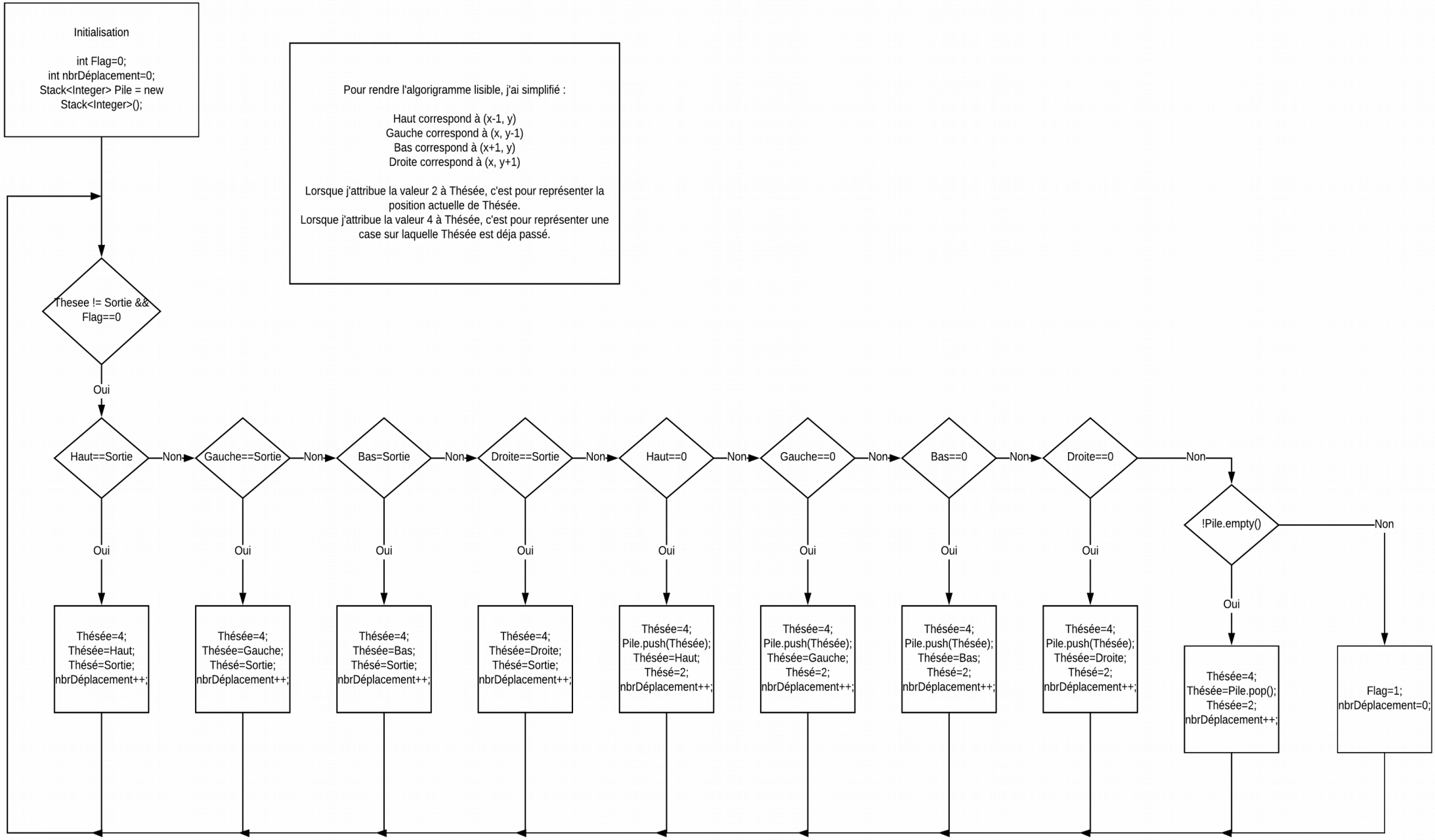


Diagramme de classes

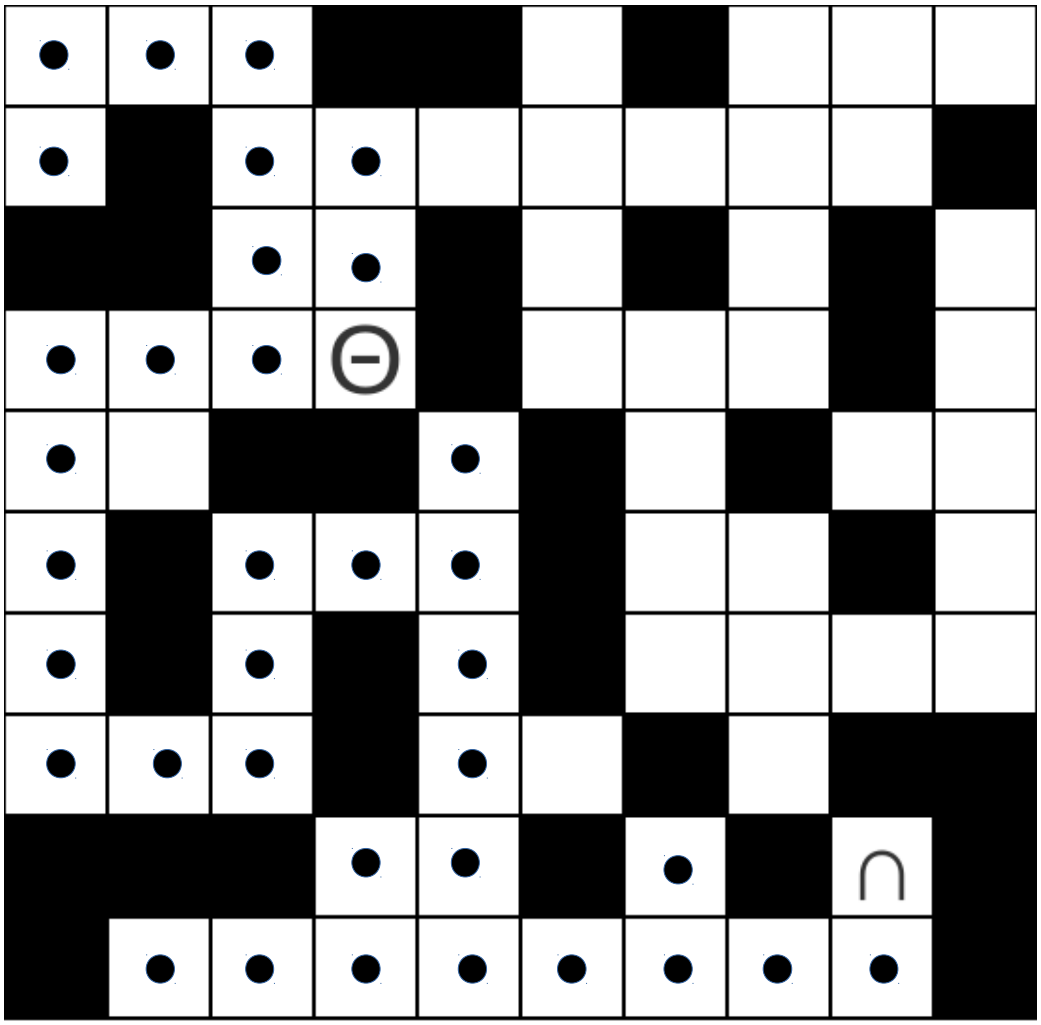


Exposition de l’algorithme déterministe

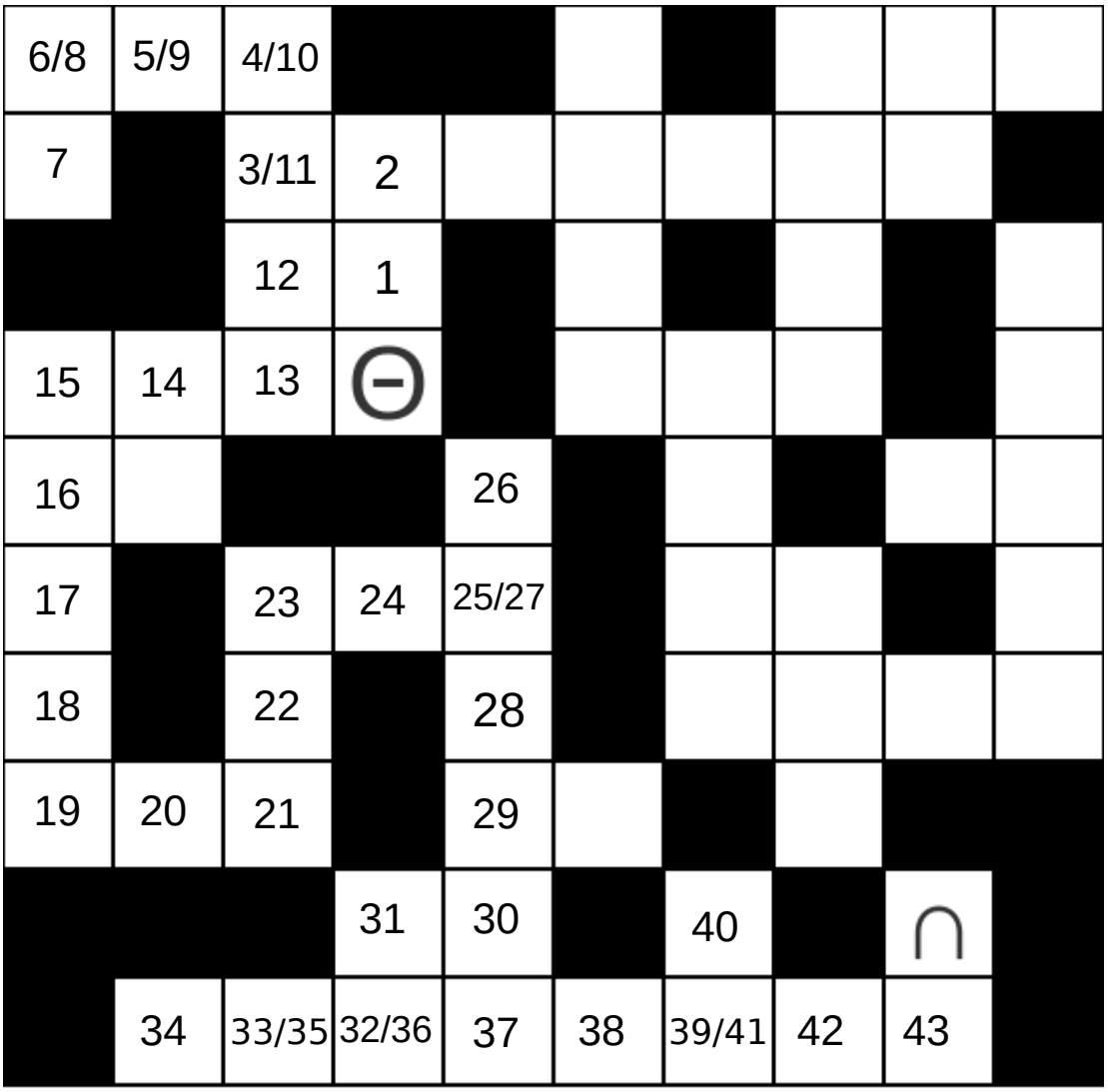


Pour la démonstration de l'algorithme déterministe, prenons par exemple le labyrinthe du sujet. Notre algorithme exécute toujours le même ordre de direction : haut, gauche, bas, droite.

Tout d'abord, la première étape pour Thésée consiste à observer si la sortie se trouve autour de lui. Si la sortie se trouve près de lui, alors il y va. Sinon il regarde si il y a une case près de lui qui est libre (c'est-à-dire égale à 0). Dans le cas où la case est libre, alors il y va, tout en plaçant un marqueur sur l'ancienne case (par exemple 4 dans notre programme) et en empilant les coordonnées de l'ancienne position dans une pile. Ainsi Thésée pourra alors se souvenir qu'il sera déjà passé par cette case. Dans le dernier recours, où Thésée ne trouve plus de case libre autour de lui (égale à 0) alors il dépile, afin de récupérer les coordonnés de l'ancienne position et ainsi, reculer. Et cela seulement, si la pile n'est pas vide. Dans le cas, où Thésée doit reculer mais ne peut pas car la pile est vide, cela veut donc dire qu'il aura exploré toutes les cases sans jamais trouvé la sortie et que donc le labyrinthe, dans lequel il se trouve, est impossible à résoudre. Cette série d'actions est réalisé jusqu'à que Thésée atteigne la sortie ou qu'il ait exploré toutes les cases du labyrinthe.



Voici en détail, la décomposition des déplacements de Thésée qui montre sa progression dans le labyrinthe. Ce labyrinthe a donc été résolu par l'algorithme déterministe avec un total de 44 déplacements.



Conclusion personnelle Nicolas FAFIN

La mise en œuvre de ce projet m'a permis d'approfondir mes compétences en JAVA et d'acquérir de nouvelles méthodes de travail.

En effet, travailler en duo avec Antoine a pu m'apporter de nombreuses satisfactions. Chacun de nous deux a pu apprendre d'avantage sur les éléments qui constituaient un groupe : ses méthodes de travail, son organisation, son esprit d'analyse et surtout sa communication.

Ce projet a donc été une expérience enrichissante où l'autonomie, le sérieux et le travail de groupe était présent et nécessaire.
Partager ses connaissances et débattre autour de ce projet a réellement été constructif pour moi.

Je remercie Luc Hernandez pour ses réponses aux diverses questions que j'ai pu lui envoyer.

Pour finir, je suis vraiment content du rendu final de notre projet.
Collaborer avec Antoine était un réel plaisir et la réalisation de ce programme est l'une des meilleures expériences que j'ai pu avoir jusqu'à présent.

Ce projet en langage java m'a vraiment plu et me servira pour la suite de mes études.

Conclusion personnelle Antoine MAN

J’ai adoré réaliser ce projet en JAVA car il nous a permit de revoir tout ce que l’on avait appris jusqu’à là, et ainsi confirmer nos compétences.
De plus, j’ai trouvé le sujet très intéressant avec le petit contexte mythologique.

La réalisation de ce projet avec Nicolas a été une excellente expérience que je n’hésiterai pas à refaire, car j’ai pu rencontrer une personne différente, quant il s’agit de travailler en équipe, à la fois rigoureux, assidu et motivé.

En effet, ce projet nous a permit de développer chacun de notre coté, des méthodes de travail en équipe telles que l’organisation ou encore la communication.

Nous avons fait preuve de sérieux tout au long de la réalisation de ce projet, qui peut notamment se refléter à travers le résultat obtenu. En effet, je suis très satisfait du travail que nous avons pu fournir et du résultat final.