

Rapport de SAE : Jeu de paires

Par Côme Thuret et Axel Pietrois

Sommaire :

1. Introduction
2. Fonctionnalités du programme
3. Structure du programme
4. Conclusions personnelles

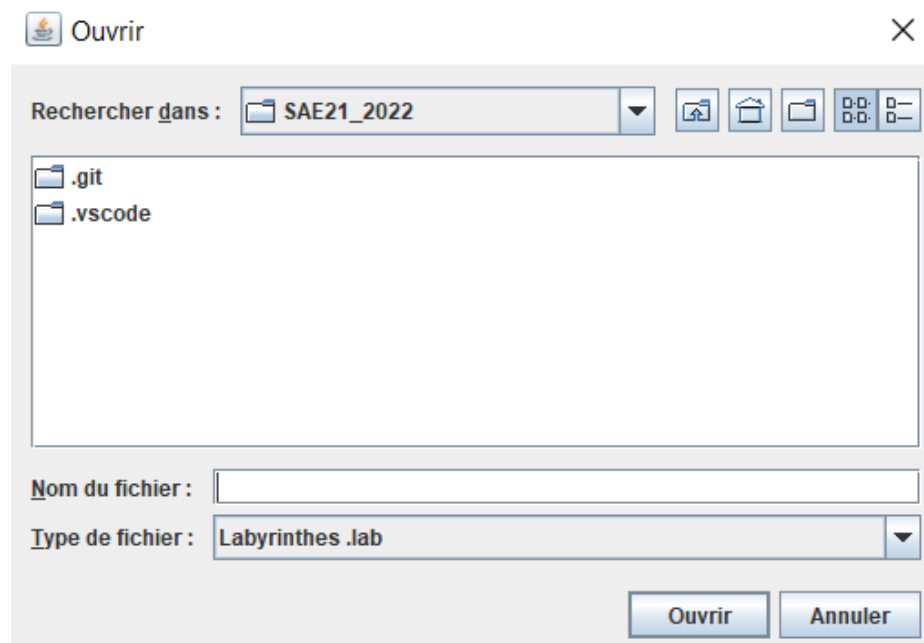
1. Introduction

Ici le travail demandé est de réaliser un programme en Java ayant plusieurs fonctionnalités différentes. l'utilisateur doit tout d'abord avoir la possibilité de charger un labyrinthe à partir d'un fichier, ou bien d'en générer un. Pour la génération l'utilisateur aura la possibilité de choisir entre une génération aléatoire ou bien de créer un labyrinthe vide , tous deux selon une taille donnée. L'étape d'après est la modification du labyrinthe , en effet lors de cette dernière, l'utilisateur pourra modifier la structure du labyrinthe (changer la position des murs) ainsi que de donner le point de départ de Thésée et la sortie du labyrinthe. Une fois cela fait, l'utilisateur pourra sauvegarder ou non le labyrinthe dans un fichier, puis choisir un algorithme qui résoudra le labyrinthe parmi les suivants : algorithme aléatoire , algorithme déterministe. Ces deux algorithmes auront chacun deux fonctionnements: l'un étape par étape (dit manuel) et l'autre où l'on résout 100 fois le labyrinthe et l'on montre le nombre moyen d'étapes pour le résoudre (dit automatique).

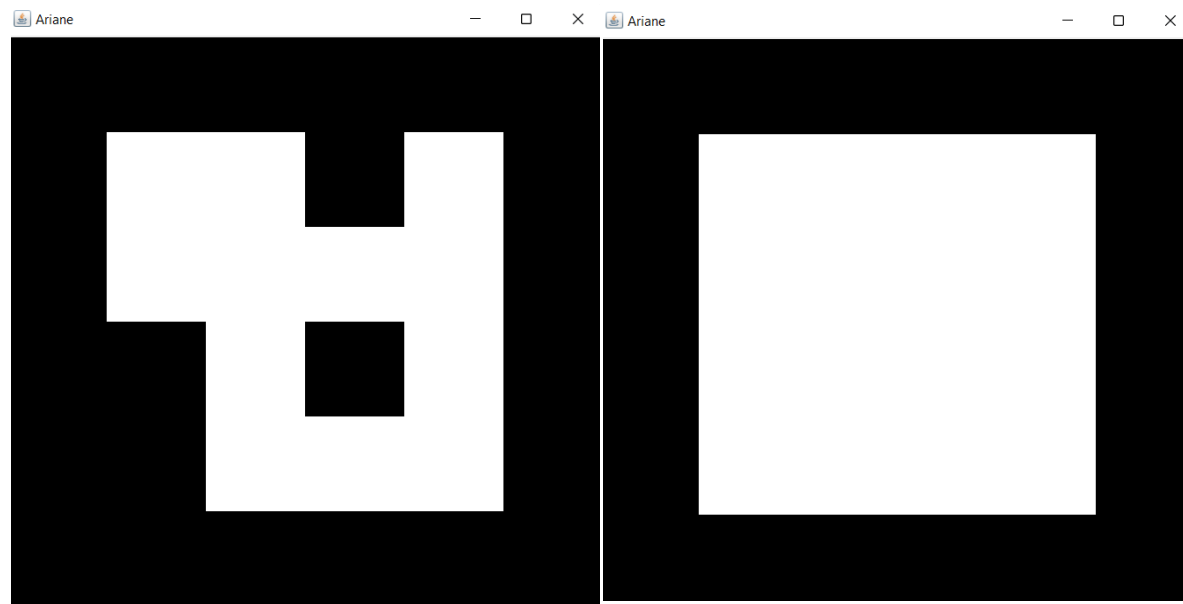
2. Fonctionnalités du programme

Ce programme contient plusieurs fonctionnalités majeures :

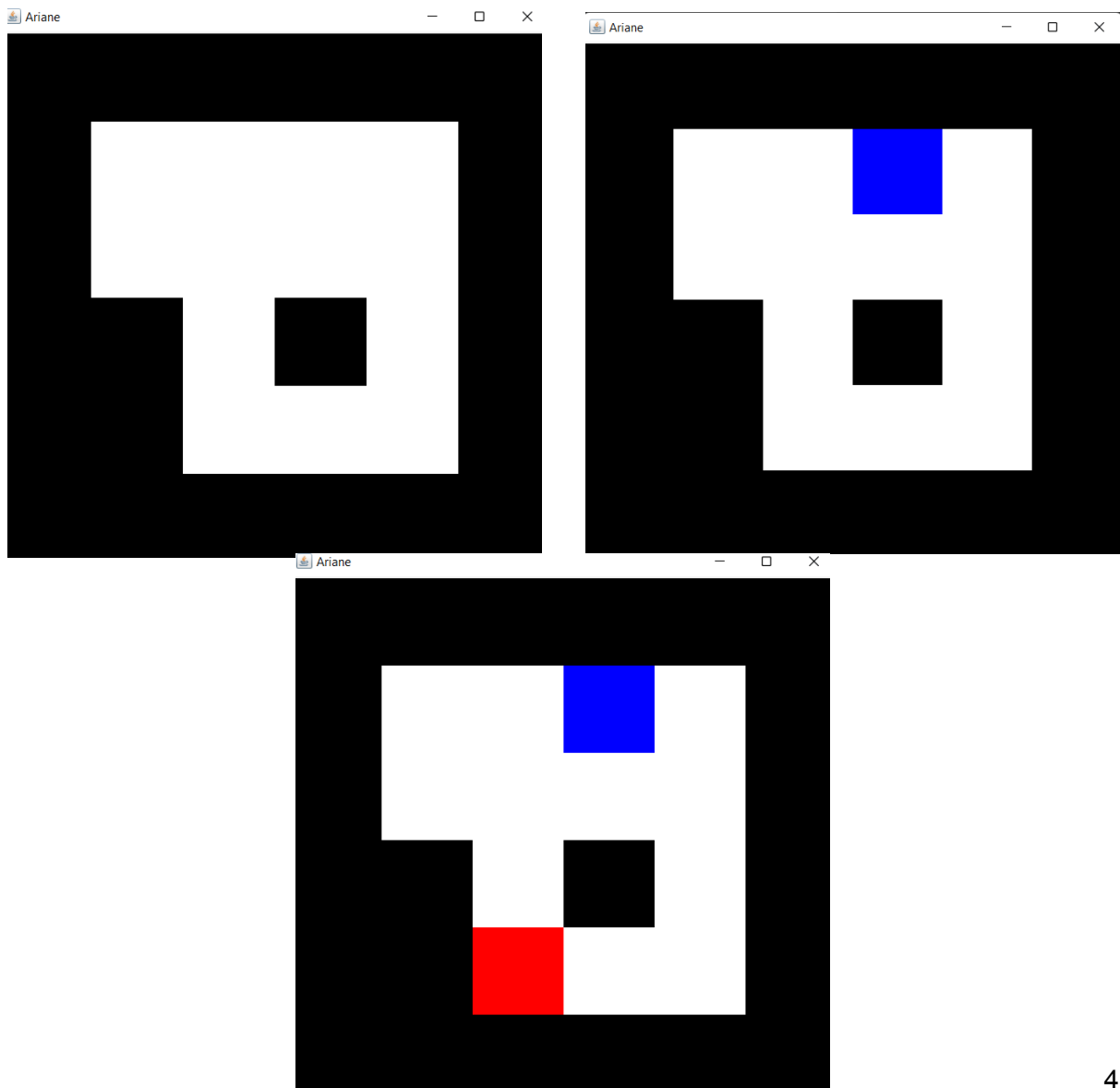
- Un menu principal permettant de: charger un labyrinthe ou d'en générer un.



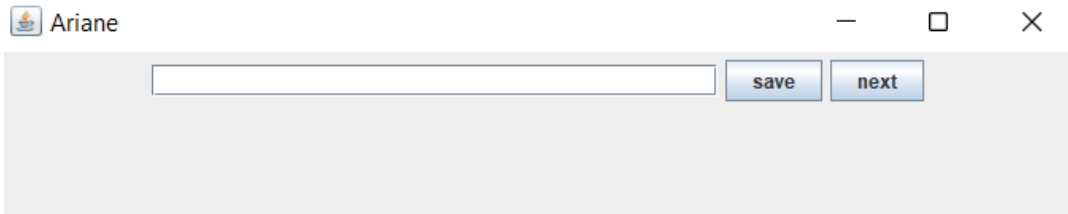
- générer un labyrinthe vide ou aléatoire



- modifier le labyrinthe (changer les murs, donner le départ en bleu et la fin en rouge) et appuyer sur espace pour arrêter la modification.



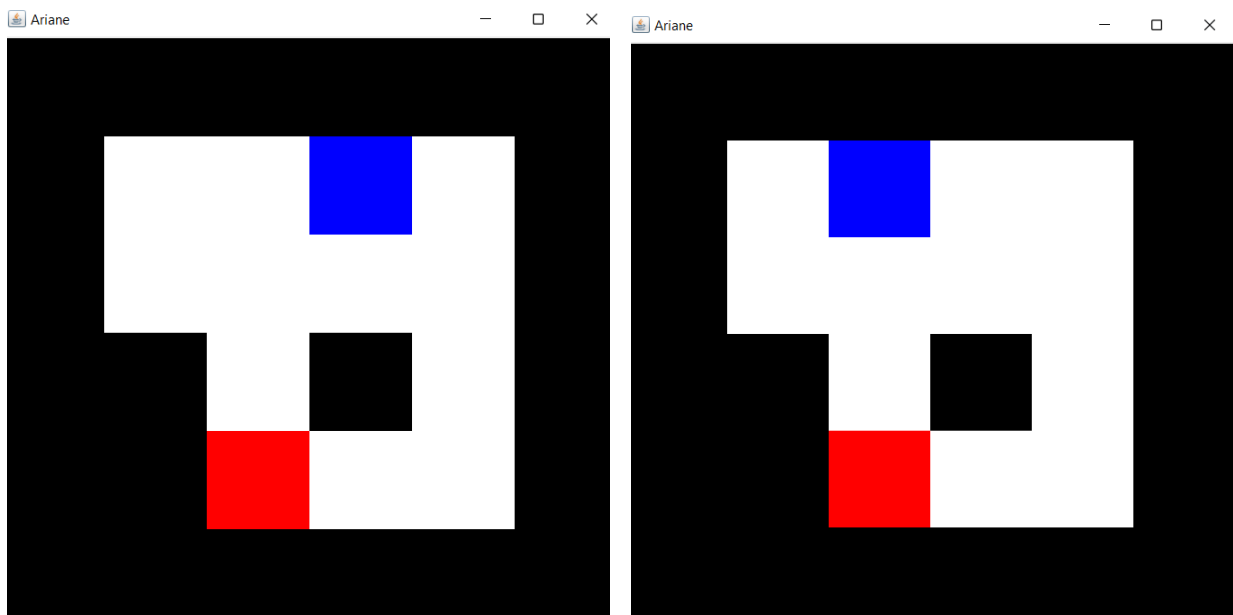
- sauvegarder le labyrinthe modifié précédemment



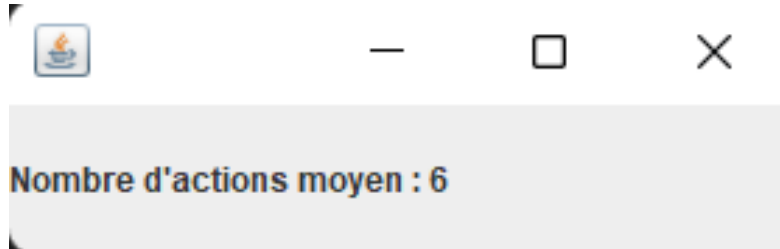
- Choisir entre un algorithme déterministe (automatique) et un algorithme aléatoire (manuel)



- Voir les déplacements de l'algorithme aléatoire étape par étape en appuyant sur la barre espace.



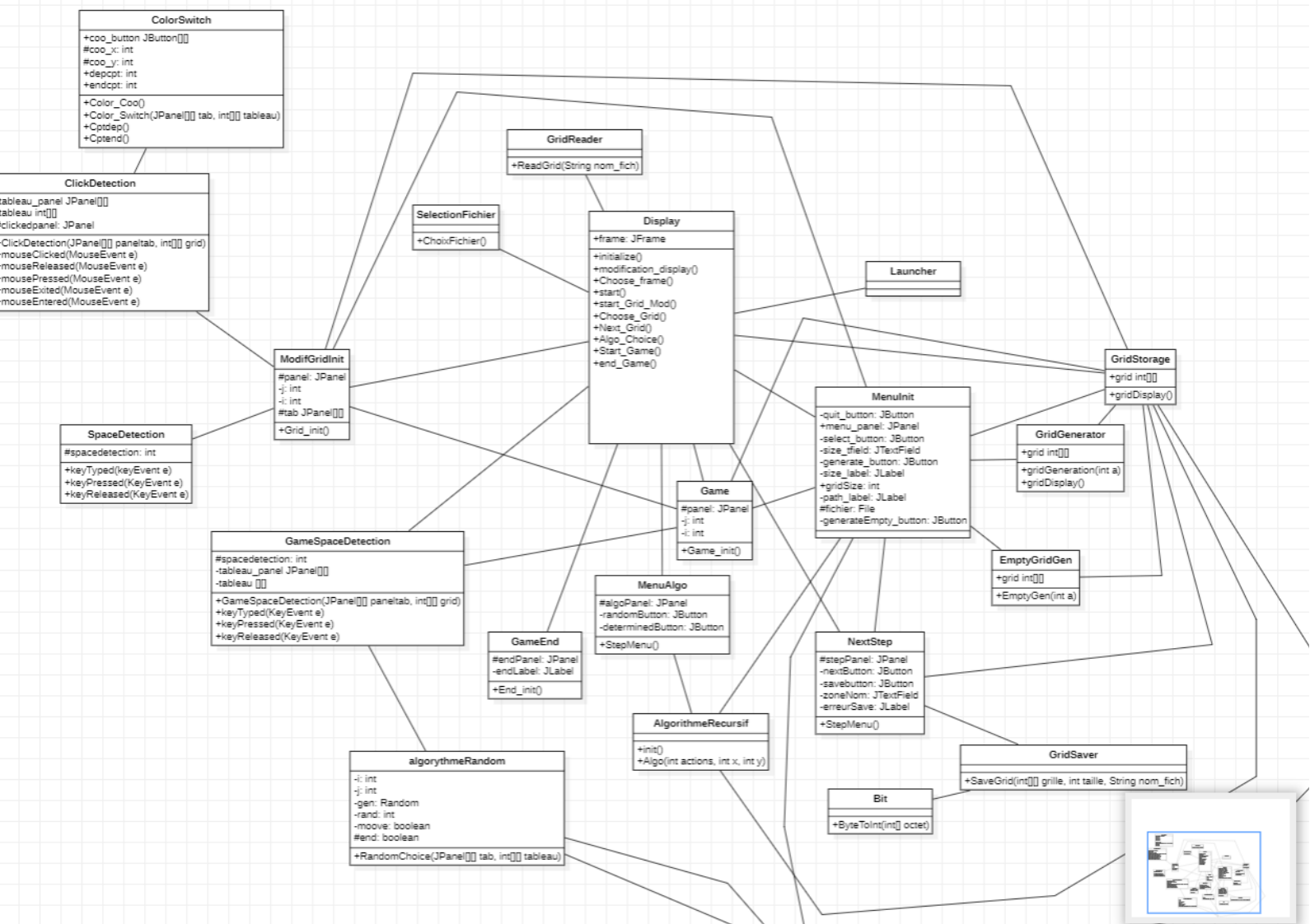
- Voir le nombre de déplacement moyen de l'algorithme déterministe



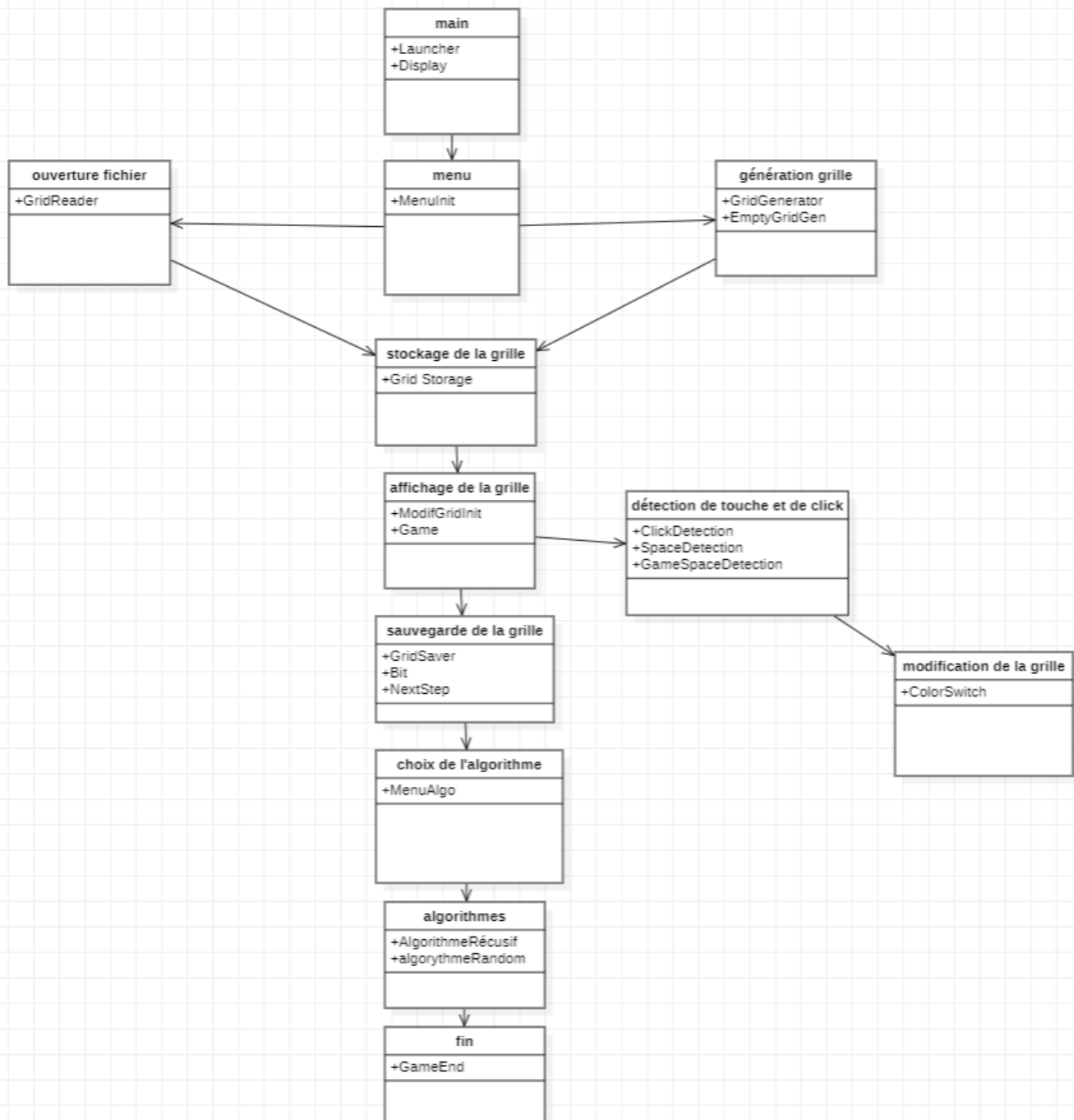
3. Structure du programme

L'entièreté du programme est divisée en plusieurs fichiers, 21 pour être précis. Et chacun de ces fichiers comporte une classe qui elle-même comporte une ou plusieurs méthodes.

- diagramme complet :



- diagramme de fonction des classes :



4. Conclusions personnelles

Axel Pietrois :

Pour ma part, ce projet a été enrichissant car je me suis occupé de points que je n'avais pas énormément utilisés avant. Cependant, l'organisation était assez mitigée, au début du projet, j'ai eu un élan de motivation, que j'ai perdu assez vite, ce qui fait que j'écris ces lignes à 22h16, c'est-à-dire moins de deux heures avant la date de rendu. Je pense que la fin était assez difficile, et cela peut se lire dans nos commits. Le projet aurait pu être bien mieux fini avec une meilleure organisation au niveau du temps (ou une semaine en plus), je ne suis pas réellement satisfait du travail rendu, mais il fallait rendre quelque chose, et on a préféré rendre quelque chose avec le maximum de fonctionnalités un peu bancales plutôt que de rendre un projet qui ne fonctionne pas mais les quelques fonctionnalités existantes soient sans problèmes. Pour conclure, ce projet aurait pu mieux se dérouler, mais c'est ce genre de projet qui nous donne de l'expérience et qui nous empêchera de refaire ce genre d'erreur à l'avenir.

Côme Thuret : h

Pour ma part il y a plusieurs points importants à aborder. Tout d'abord il y a ce qui aurait pu être amélioré comme bien évidemment la gestion du temps, car selon moi notre temps n'as pa été assez bien gérée et nous à fait travailler dans la "précipitation"; à cette gestion de temps, je pense que nous n'avons pas bien défini les priorités de développement, j'entend par là que nous avons développé certaines fonctions dans le mauvais ordre ce qui à pu augmenter notre retard. Cependant les fonctionnalités que nous avons implémentées sont toutes fonctionnelles et je trouve à titre personnel que le tout fonctionne bien, même si de nombreuses améliorations pourraient être effectuées.