

Rapport de Projet JAVA :
CY-Slide



**Andrew Bocq - Julian Gosselin - Théo Grandjean -
Paul Jumel - Anatole Paques**

2022-2023
ING1 - GI1

SOMMAIRE

- I. INTRODUCTION
- II. REPARTITION DES TACHES ET ORGANISATION
- III. PLANNING
- IV. SCHEMA USE CASE, DIAGRAMME DE CLASSE ET DE SEQUENCE
- V. PROBLEMATIQUES ET SOLUTIONS / AVANTAGES ET LIMITES / AMELIORATIONS
- VI. CONCLUSION

Introduction

Le présent rapport de projet porte sur la réalisation du jeu de taquin en Java. Ce jeu classique consiste à reconstituer une image ou un motif en déplaçant des pièces numérotées sur une grille jusqu'à ce qu'elles soient dans l'ordre correct.

L'application de jeu de taquin a été développée en utilisant Java et la plate-forme JavaFX pour l'interface graphique. Le projet a été structuré en plusieurs classes et modules, permettant une séparation claire des différents organes du projet.

Ce rapport présentera une vue d'ensemble du projet, l'organisation ainsi que les défis rencontrés et les solutions adoptées. Il fournira également des perspectives d'amélioration et les leçons apprises tout au long du processus de développement.

La suite du rapport décrira plus en détail les différentes fonctionnalités du jeu de taquin, les problèmes techniques rencontrés, ainsi que les résultats obtenus.

Répartition des tâches et organisation

Pour la répartition des tâches, nous avons chacun choisi la partie qui nous intéressait le plus. Ainsi, Paul s'est occupé de réaliser le jeu en lui-même, la jouabilité et l'affichage du jeu. Théo a réalisé l'interface graphique du jeu, il a ainsi créé le menu à gauche et rassemblé les deux parties sur une scène et les fonctions qui permettent à l'interface et au jeu de s'actualiser au fur et à mesure du jeu. Julian a pour sa part géré la gestion de niveau et la lecture de fichier, nous retrouvons dans sa partie la création des différents niveaux, l'initialisation du jeu avec le fichier niveau, le stockage du score le plus faible. Anatole s'est occupé de la partie solution automatique, il a réalisé un petit algorithme qui permet de trouver la solution et de résoudre le niveau actuel. Pour finir, Andrew s'est occupé de détecter l'existence d'une solution après un mélange.

Nous avons utilisé différents types d'outils lors de la réalisation de ce projet. Tout d'abord, nous avons utilisé Eclipse comme IDE pour développer avec le langage Java. Nous avons également utilisé GitHub pour le partage de code et de fichier et Discord comme logiciel de communication.

Pour réaliser ce projet, nous avons travaillé de manière régulière afin de ne pas être submergé par la masse de travail dû à un potentiel retard. Le travail fourni était sérieux et à chaque session virtuelle nous avançons de manière considérable. En effet, dans un premier temps nous avons fixé un rendez-vous tous les deux jours, mais lors de la seconde semaine nous avons décidé de faire un rendez-vous quotidien de 14h à 18h, cela nous a permis de garder un rythme de travail efficace et de créer une bonne cohésion d'équipe. Si on avait des questions ou des problématiques en rapport avec d'autre partie, nous pouvions directement demander à l'individu concerné, cela nous a permis d'avancer plus vite.

Planning

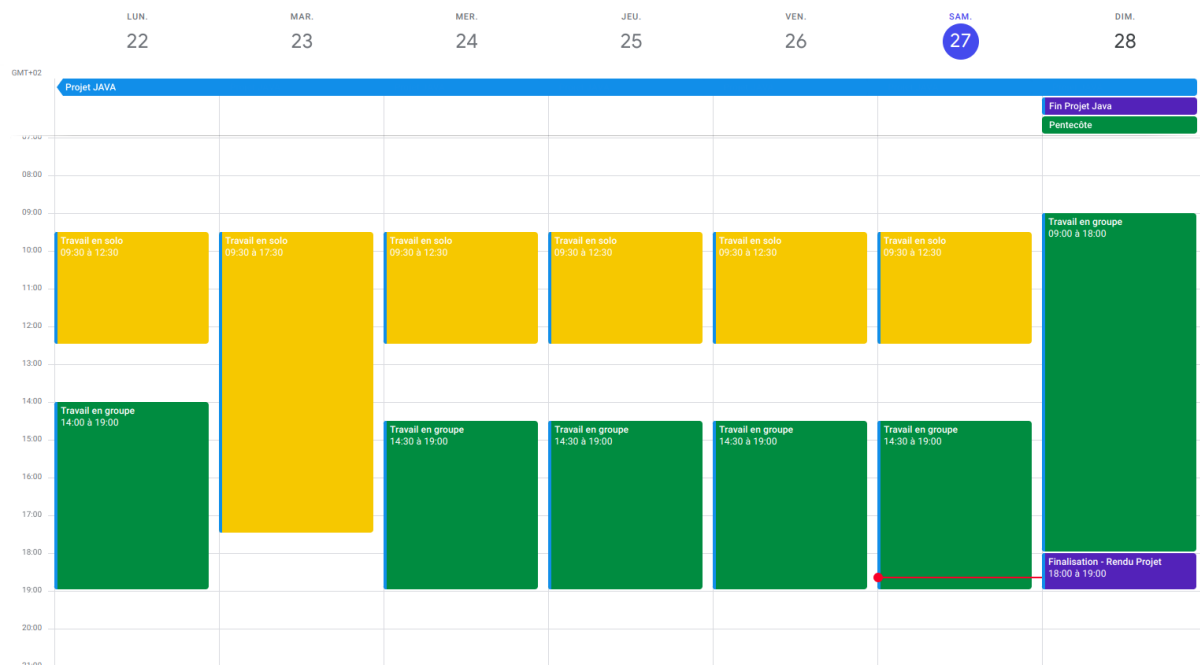
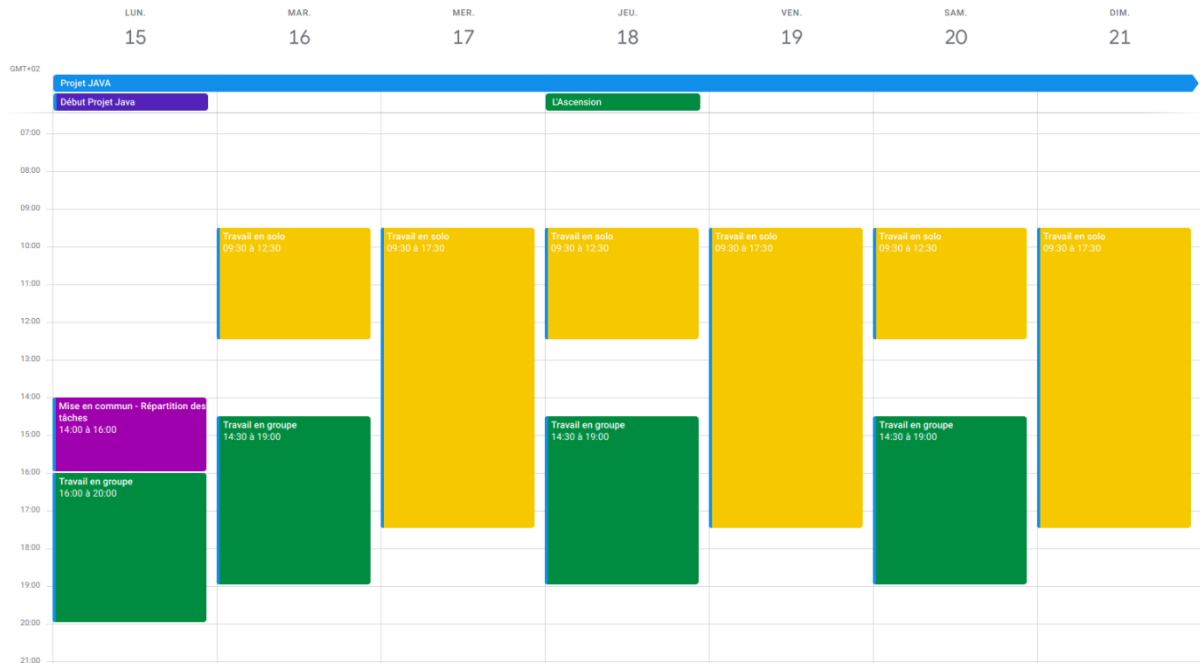


Schéma use case, Diagramme de classe et de séquence

Schéma use case :

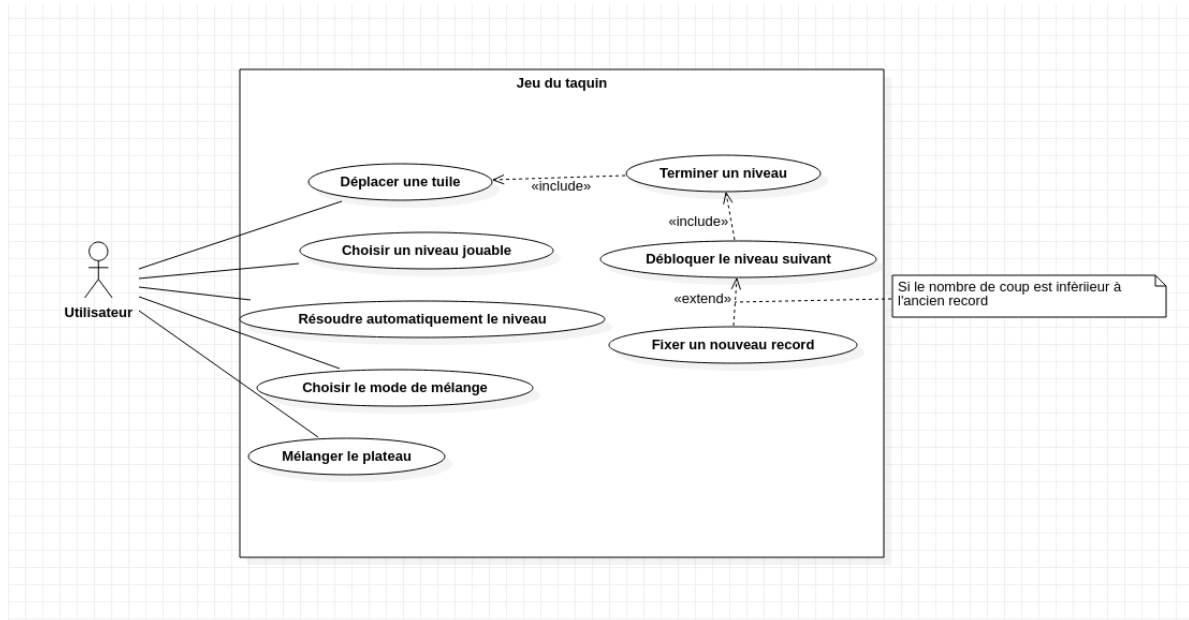


Diagramme de classe :

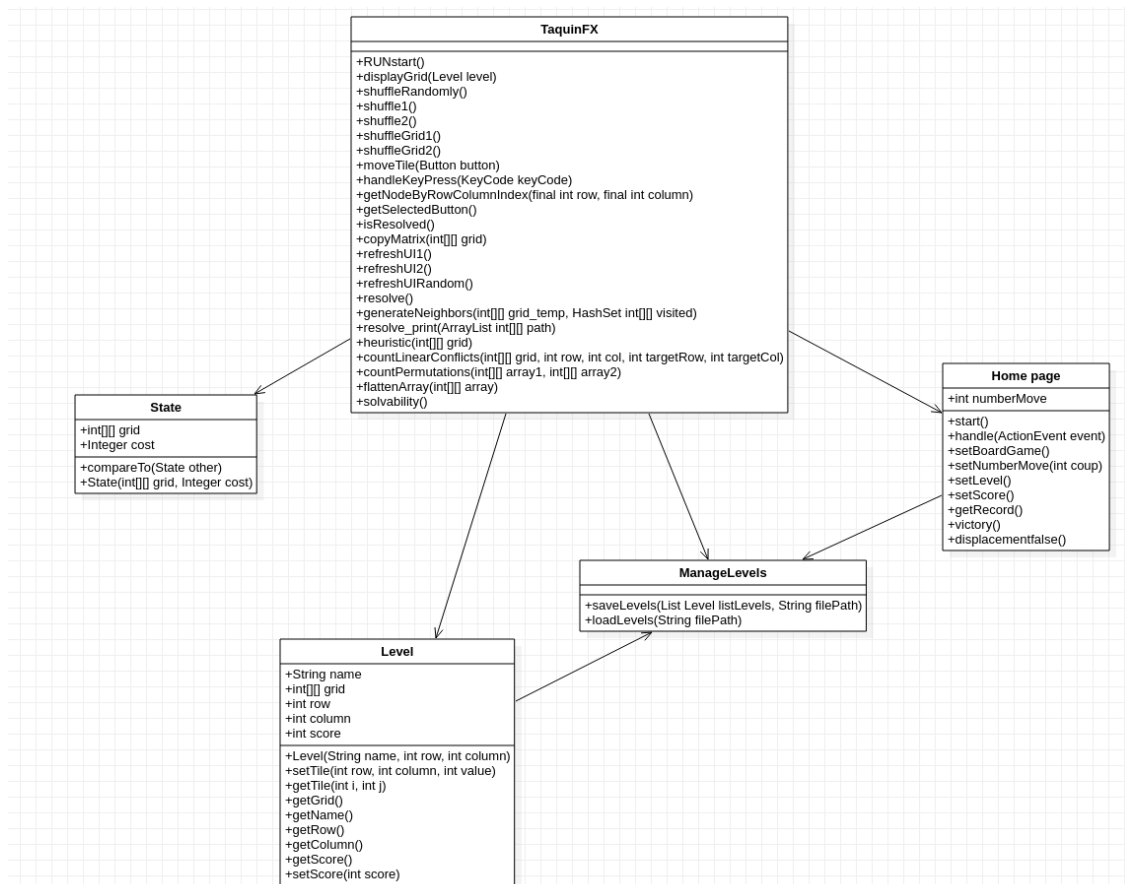
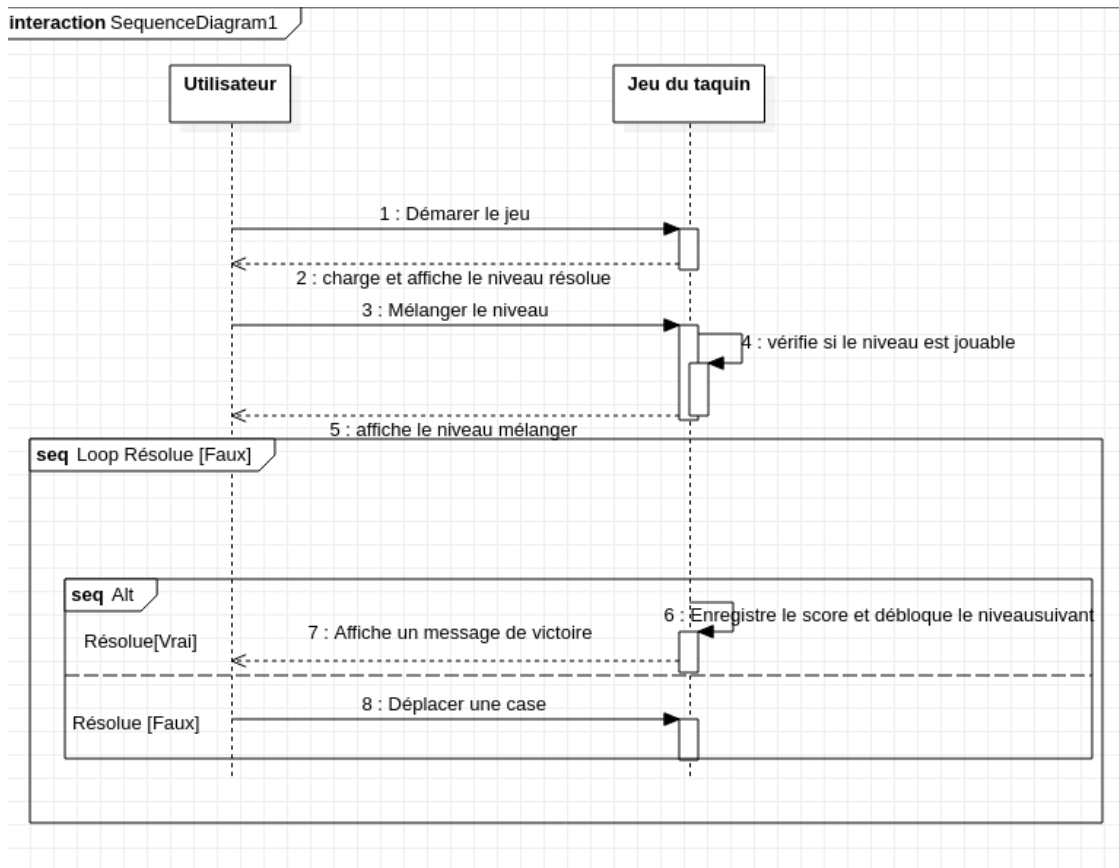


Diagramme de séquence :



Problématiques et Solutions / Avantages et limites / Améliorations

Problématiques et Solutions :

Dans ce projet nous n'avons pas rencontré beaucoup de problèmes majeurs, tout s'est plutôt bien déroulé et nous trouvions toujours les solutions très rapidement.

Seul le problème de la résolution automatique persiste, bien qu'elle fonctionne, elle prend un temps considérable et trop d'espace mémoire lorsque le jeu est beaucoup mélangé ou lorsque les dimensions du jeu sont trop importantes. Pour ce faire nous avons essayé différents types d'algorithmes comme l'algorithme A* ou IDA*, de plus, nous avons essayé de les optimiser en améliorant la fonction d'évaluation (utilisation de la distance de Manhattan, conflits linéaires, ...) ou en ajoutant une file de priorité pour gérer les états du jeu.

Un autre souci était qu'au commencement nous avons créé le menu et le jeu sur deux fenêtres séparées et nous ne savions pas vraiment si cela était en accord avec le cahier des charges. De plus, cela pouvait causer des problèmes par rapport aux interactions, à l'actualisation du menu et du jeu. Nous avons donc regroupé les deux fenêtres en utilisant l'option SplitPane de JavaFx.

Avantages :

Notre projet possède beaucoup d'avantages, il respecte avant tout le cahier des charges dans son intégralité, mais nous avons aussi :

- La possibilité d'afficher les règles du jeu,
- Le joueur peut retenter les niveaux autant de fois qu'il le veut
- Lorsque le joueur gagne le plateau devient vert, si la case ne peut pas être déplacé et qu'il clique dessus, alors un message rouge s'affiche pendant 1 seconde.
- Le jeu et le menu sont sur la même fenêtre et tout s'actualise en temps réel sur la fenêtre.

Limites :

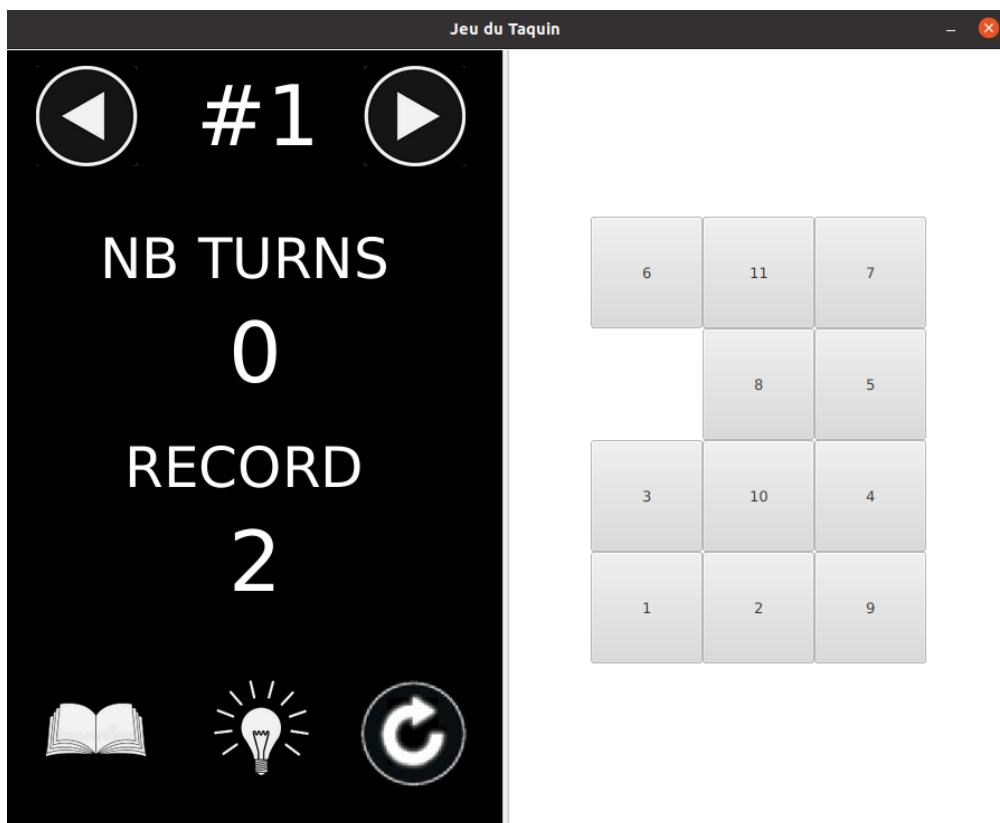
Notre jeu ne possède que 2 limites :

- On ne peut avoir qu'une seule case vide jouable
- La résolution automatique prend beaucoup trop de temps lorsque le jeu est beaucoup mélangé ou lorsque les dimensions du jeu sont trop importantes (supérieur à 4x4).

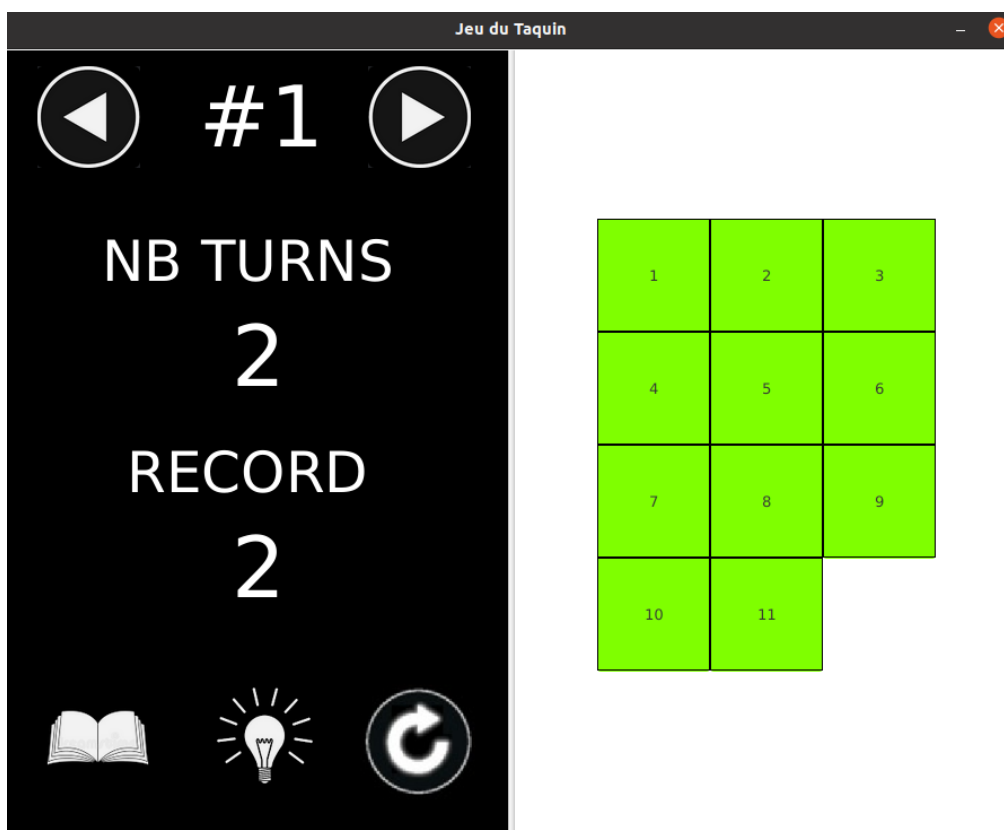
Amélioration :

- Il faudrait mettre la possibilité de pouvoir choisir pour chaque niveau le nombre de case vide qu'on veut et les implémenter dans le jeu, ainsi que dans les critères de la résolution automatique
- Rendre la résolution automatique plus rapide et faire en sorte qu'elle ne prenne pas trop de mémoire.

Jeu en général :



Animation pour la résolution :



Conclusion

Pour conclure, ce projet nous a permis de nous familiariser avec JAVA et Eclipse. Pour la première fois, nous avons créé une interface graphique fonctionnelle car jusque-là nous n'avions fait que des programmes avec un affichage dans le terminal. Nous avons également pu adapter des algorithmes vus en cours, afin de développer la solution automatique. Nous avons amélioré nos compétences de travail en équipe en s'organisant à l'avance et en mettant en commun nos avancées sur la plateforme GitHub. De manière générale, ce projet fut une expérience enrichissante qui nous a permis d'améliorer nos compétences en programmation.

Andrew, Anatole, Paul, Théo, Julian.

ING 1 Groupe 1.

2022-2023.