

# PROJET GENIE LOGICIEL : Rapport final

---

Sujet : CY-Slide

Professeur : M. Julien MERCADAL

28 MAI 2023

---

AÏT CHADI Anissa

BERGERE Sarah

COSTA Mathéo

FELGINES Sara

ING1 GI Groupe 4



## TABLE DES MATIERES

<b>Introduction.....</b>	<b>3</b>
<b>Organisation .....</b>	<b>4</b>
Méthode de travail .....	4
Rythme de travail et outils .....	4
<b>Planning détaillé.....</b>	<b>6</b>
<b>Problèmes rencontrés .....</b>	<b>8</b>
Incompréhensions du sujet .....	8
Problématiques sur l’algorithmie .....	8
<b>Répartition des tâches .....</b>	<b>10</b>
<b>Fonctionnalités.....</b>	<b>11</b>
Fonctionnalités supplémentaires .....	11
Axes d’amélioration .....	11
<b>Diagrammes.....</b>	<b>12</b>
<b>Conclusion.....</b>	<b>14</b>

# INTRODUCTION

L'objectif de ce rapport est d'expliquer comment s'est déroulé le projet génie logiciel. Pour cela, nous allons détailler comment nous nous sommes organisés, les divers problèmes rencontrés et comment nous les avons surmontés. Ensuite, nous évoquerons la répartition des tâches ainsi que les fonctionnalités supplémentaires que nous avons décidé d'implémenter.

Avant d'entrer dans le vif du sujet, nous allons rapidement parler des phases préliminaires du projet.

Notre équipe s'est formée très rapidement. En effet, nous avons déjà eu l'occasion de travailler ensemble à plusieurs reprises et sur divers projets. Il allait donc de soi que nous allions continuer sur cette lancée.

Concernant le choix du sujet, nous avons mis quelque temps à nous décider. Au premier abord, les sujets ne nous paraissaient pas très intéressants. Etant donné que nous nous attendions à un jeu de type quiz, nous avons été déconcertés lorsqu'il n'est pas apparu lors de la réunion de présentation. C'est donc en lisant les différents cahiers des charges que nous avons décidé de choisir le taquin, qui paraissait le plus amusant. Dès lors que nous avons choisi notre sujet, nous avons créé un groupe sur Snapchat pour communiquer rapidement ainsi qu'un Github pour y déposer nos fichiers.

# ORGANISATION

Nous avons débuté par une première réunion de lancement avec notre professeur référent, le lundi 15 mai, pour poser les bases. Etant donné que Mathéo, particulièrement intéressé par ce sujet, avait déjà commencé le projet durant le week-end, et que ce qu'il avait fait convenait aux autres, nous avons décidé de continuer sur cette base, sans penser au fait que ce n'était pas la bonne façon de faire.

C'est lors de notre deuxième réunion avec M. Mercadal, que nous nous sommes rendus compte que nous devions totalement revoir notre modèle de données et que nous avons donc décidé de tout recommencer. Cela a été très bénéfique puisque nous avons eu un code beaucoup plus clair et modulable.

## ■ METHODE DE TRAVAIL :

Concernant la méthode de travail, nous avons décidé de commencer tous ensemble. C'est-à-dire que nous nous sommes réunis à l'école, puis via un appel teams pour établir notre modèle de données, les classes dont nous allions avoir besoin, ainsi que la répartition des tâches. Nous avons donc commencé à travailler sur nos parties respectives, tout en veillant à ce que tout le monde comprenne et que personne ne reste bloqué sur un problème. Pour les journées où nous n'étions pas en mesure de nous retrouver à l'école ensemble, et en cas de besoin, nous faisions des réunions teams pour mieux communiquer sur l'avancée de nos parties et réfléchir à des solutions ensemble (notamment sur la partie algorithmie, sur laquelle nous reviendrons un peu plus tard dans ce rapport).

## ■ RYTHME DE TRAVAIL ET OUTILS UTILISES :

Pour le rythme, nous n'avons pas voulu nous fixer de volume horaire, dans la mesure où il pouvait y avoir des jours où nous travaillions plus ou moins que prévu. Nous avons décidé de travailler tous les jours, y compris le week-end pour pouvoir avancer au mieux. Certains travaillaient plus efficacement le matin, d'autres l'après-midi ou encore en soirée. La finalité était que nous devions avancer et rendre quelque chose de fonctionnel le soir.

Concernant les outils utilisés, nous avons choisi Snapchat et teams comme moyen de communication, Github pour le dépôt de fichiers et Chatgpt comme cinquième membre.

# PLANNING DETAILLE

## Semaine 1 :

### **LUNDI 15/05 :**

- Réunion de lancement avec M. MERCADAL
- Poursuite du projet déjà commencé, ensemble à l'école
- Poursuite du projet

### **MARDI 16/05 :**

- Réunion avec M. MERCADAL pour évoquer les problèmes de notre méthode de travail/ce que nous avons déjà fait
- Appel teams pour repartir de 0
- Création du nouveau modèle de données
- Création des premières classes « principales » + du plateau via lecture du csv

### **MERCREDI 17/05 :**

- Optimisation des classes
- Réflexion sur la partie algorithmique
- Ajout du solveur en utilisant l'algorithme A\*
- Choix de la case vide lorsqu'il y en a plusieurs

### **JEUDI 18/05 :**

- Rajout de nouveaux niveaux pour tester
- Compteur de coups + message de victoire
- Menu pour sélectionner le niveau de jeu

### **VENDREDI 19/05 :**

- Gestion de la sauvegarde (= niveau accessible ou non, meilleur score)
- Chronomètre
- Optimisation sauvegarde des meilleurs scores
- Algorithme A\*

### **SAMEDI 20/05 :**

- Optimisation du chronomètre
- Optimisation de la sauvegarde
- Meilleur temps sauvegardé
- Réflexions sur l'algorithme

### **DIMANCHE 21/05 :**

- Affichage des niveaux dans l'ordre
- Réflexions sur l'algorithme

## Semaine 2 :

### **LUNDI 22/05 :**

- Réunion avec M. MERCADAL
- Réflexions sur l'algorithme
- Optimisation de la sauvegarde
- Ajout de nouveaux niveaux

### **MARDI 23/05 :**

- Carte de jeu (IHM)
- Optimisation de l'algorithme
- Algorithme IDA\*

### **MERCREDI 24/05 :**

- Menu (IHM)
- Plateau de jeu

### **JEUDI 25/05 :**

- Ajout de la musique de fond
- Css
- Plateau de jeu

### **VENDREDI 26/05 :**

- Css
- Plateau de jeu

### **SAMEDI 27/05 :**

- Css
- Ajout son boutons
- Optimisation plateau de jeu

### **DIMANCHE 28/05 :**

- Dernières modifications
- Css
- Meilleurs plateaux de jeu

## PROBLEMES RENCONTRES

Nous avons dû faire face à plusieurs problèmes durant notre projet et le plus gros a été la partie algorithmique, notamment pour le solveur et la vérification concernant si un niveau est faisable ou non.

### ■ INCOMPREHENSIONS DU SUJET :

Nous n'avions pas compris la tâche « Afficher un message si un niveau est jouable ou non (si le joueur peut arriver à le terminer) ». Nous pensions au départ que si un niveau n'était pas faisable, il fallait afficher un message et le joueur ne pouvait pas réussir le niveau. Mais, cela était en contradiction avec le fait qu'il fallait débloquer chaque niveau pour pouvoir passer au suivant. Nous avons donc décidé de diviser les cas selon la « faisabilité » du niveau. C'est-à-dire que nous proposons au joueur seulement des niveaux résolubles. Mais, si un niveau a pris trop de temps à être résolu par le solveur ou bien que la configuration n'est pas bonne, un message s'affiche dans la console.

### ■ PROBLEMATIQUES SUR L'ALGORITHMIE :

La première difficulté a été de trouver l'algorithme le plus optimal pour les différentes configurations (avec plusieurs cases vides, avec des blocs...). En effet, il nous fallait un algorithme de recherche rapide et qui n'utilisait pas trop de mémoire. Nous avons donc débuté avec l'algorithme **A\***, qui permet de rechercher le meilleur chemin dans un arbre pondéré. Cette pondération se fait en fonction du coût réel du chemin parcouru depuis le premier nœud et d'une estimation heuristique du coût restant pour atteindre le nœud final.

Nous donc avons implémenté cet algorithme, mais nous avons souvent des problèmes de mémoire. Nous avons donc essayé plusieurs heuristiques pour essayer de prioriser certains chemins et donc de réduire le temps d'exécution. Cependant, même avec ces différentes heuristiques, l'algorithme prenait encore beaucoup de mémoire.

Nous avons donc fait plus de recherche et nous avons découvert l'algorithme **IDA\***, qui est une variante de A\*. Il effectue une recherche en profondeur limitée par un seuil qui est initialisé à la valeur de l'estimation heuristique du nœud de départ. Lorsque la recherche en profondeur atteint le seuil sans trouver le nœud de destination, cela signifie qu'il est nécessaire d'augmenter le seuil pour permettre une exploration plus profonde.



Le problème que nous avons eu avec cet algorithme était de gérer le cas où il existe plusieurs cases vides. En effet, au début, nous avons choisi de sélectionner une case vide de façon aléatoire sauf que cela rendait plus long le temps d'exécution car il n'y avait pas de stratégie particulière pour utiliser la "meilleure" case vide. Nous avons donc décidé de tester le déplacement de toutes les cases vides afin de ne pas en laisser de côté et donc de trouver les meilleures combinaisons possibles.

L'autre problème que nous avons rencontré est le fait que pour un même plateau de taquin et un même nombre de mélange l'algorithme pouvait prendre 2ms comme 5s ce qui n'est pas extrêmement long mais qui est quand même un grand écart entre les deux temps.

Nous n'avons malheureusement pas réussi à résoudre ce problème car nous pensons qu'il s'agit seulement de la façon dont le solveur parcourt les différents nœuds. Mais pour éviter ce problème nous avons décidé, pour tout mélange valable, de tester le mélange au préalable par le solveur et s'il prend plus de X temps alors on passe à un autre mélange. Cela assure la rapidité du solveur pour la plupart des cas.

## REPARTITION DES TACHES

TÂCHE	QUI
<b>PARTIE FONCTIONNELLE</b>	
Création du plateau	Sara, Anissa, Sarah, Mathéo
Vérifier plateau résolu + échanger cases	Sara, Mathéo, Sarah
Mélanger plateau	Sara, Mathéo
Gérer cas plusieurs cases vides	Sarah
Compteur de coups	Anissa
Choisir niveau du plateau	Anissa, Mathéo
Sauvegarde des meilleurs scores et temps	Anissa, Sarah, Mathéo
Solveur	Mathéo, Sarah
Plateau jouable ou non	Sarah
<b>IHM</b>	
Map de jeu	Sara
Affichage du plateau	Anissa
Menu	Sara
Musique de fond + sons boutons	Anissa
CSS	Sara, Anissa, Sarah
Plateau et interactions	Sarah, Mathéo, Sara

# **FONCTIONNALITES ET AXES D'AMELIORATION**

## **■ FONCTIONNALITES SUPPLEMENTAIRES**

Voici les quelques détails que nous avons voulu ajouter à notre jeu :

- Ajout d'un chronomètre pour un peu plus de challenge
- Carte de jeu comme jeux mobiles ou en ligne
- Musique de fond pour plus de réalisme

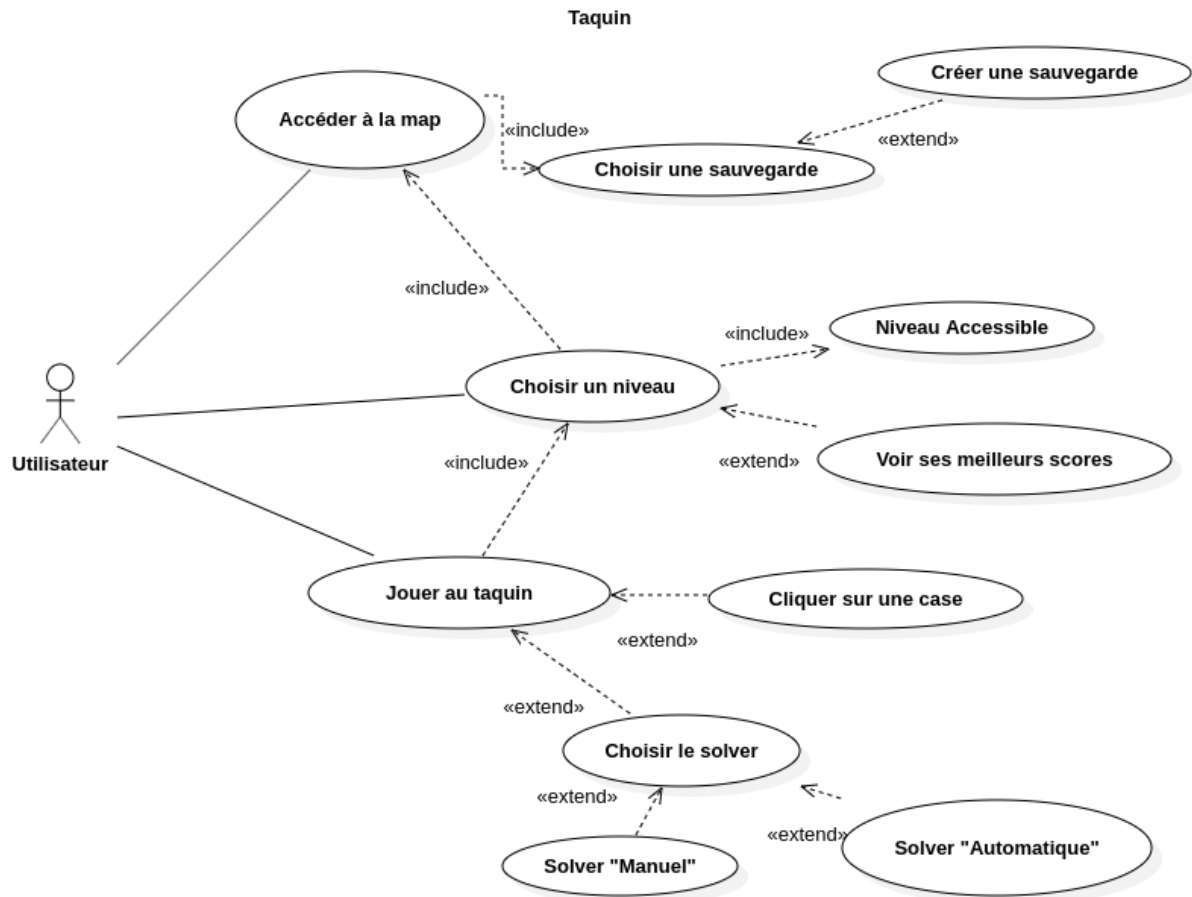
## **■ AXES D'AMELIORATION**

Nous avons songé à plusieurs fonctionnalités futures concernant notre projet :

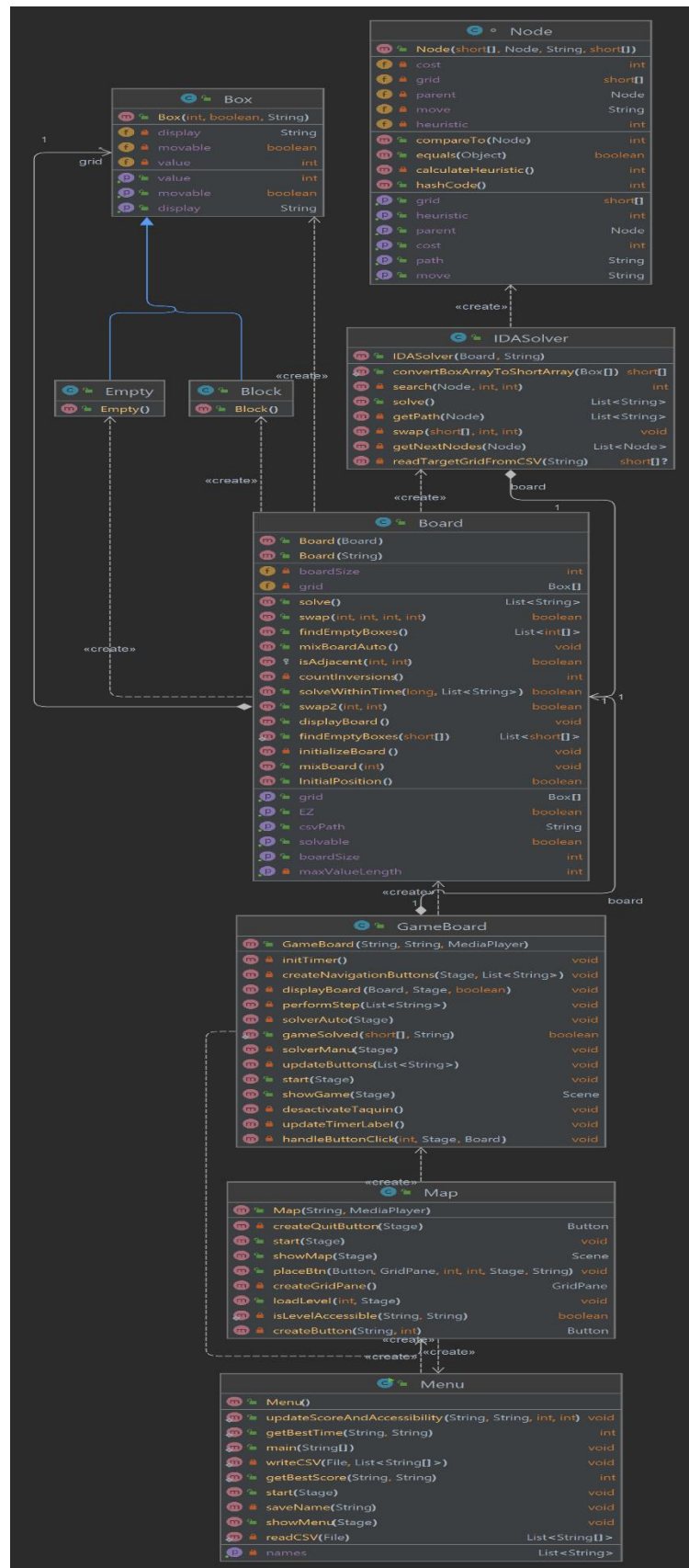
- Jouer avec une image au lieu de chiffres
- Animations
- Si jouable avec image, permettre à l'utilisateur de charger sa propre image
- Minuteur/temps imparti
- Améliorer le mélange pour qu'il se réalise dans un temps plus constant
- Trouver une autre façon de vérifier la solvabilité du jeu avec un algorithme qui prend bien en compte le fait qu'il puisse y avoir plusieurs cases vides et/ou plusieurs blocks.

# DIAGRAMMES

## ---Use Case---



## ---Diagramme de classes---



## CONCLUSION

En somme, ce projet a été l'occasion pour nous de développer nos compétences en java, ainsi que notre capacité à travailler en équipe sur un projet ayant un délai de réalisation assez court.

Très vite, nous nous sommes organisés afin de gérer au mieux la courte durée du projet.

Nous avons appris notamment à gérer le stress, les imprévus et surtout à accroître notre PATIENCE. En effet, nous avons eu beaucoup de problèmes concernant l'algorithme à implémenter, ce qui nous a un peu retardé sur notre planning d'avancée.

Nous voulions finir la partie fonctionnelle, ou du moins bien l'entamer avant de passer à la partie IHM. Comme évoqué un peu plus tôt dans ce rapport, nous avons des idées d'améliorations concernant notre projet. Nous aimerions par exemple rendre meilleur le design qui, malgré nos efforts pour le rendre attractif, manque encore de détails. Nous aimerions également pouvoir faire un minuteur, au lieu d'un chronomètre, qui, lorsque la durée est écoulée, quitte la partie en affichant « Vous avez perdu trop de temps... ».

Nous retenons cependant beaucoup de positif et nous sommes fiers de ce que nous avons réussi à réaliser en si peu de temps.