

CHARLES Ines, GUIRENE Yanis,
HAJEM Yassine, LAFON Alphonse,
LAHYEN Nouha, POIRIER Romain

Groupe AB2 : E7 et Mat, un jeu d'échecs

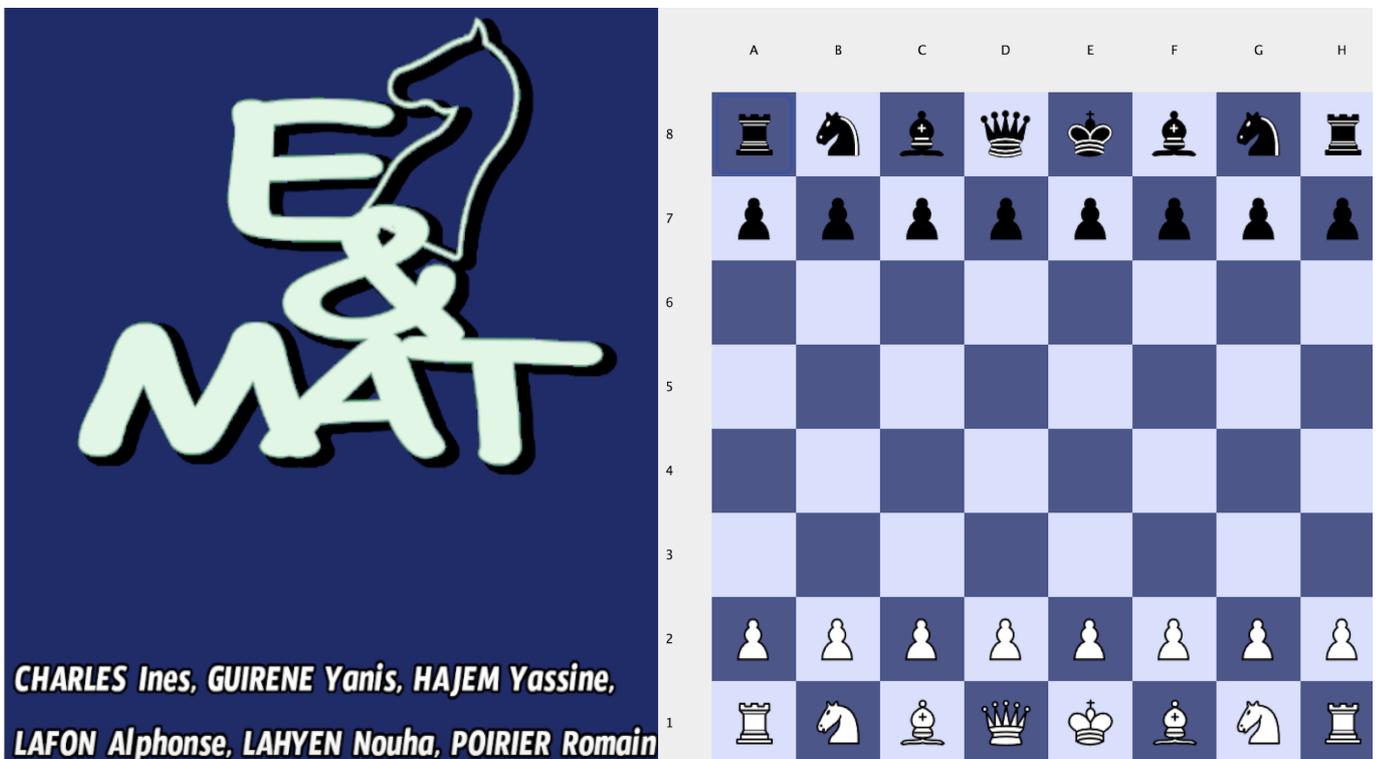


Table des matières

1	Introduction	2
2	Fonctionnalités proposées et état d'avancement	2
2.1	Fonctionnalité du jeu principal, joueur contre joueur	2
2.1.1	Itération 1	2
2.1.2	Itération 2	2
2.1.3	Itération 3	3
2.2	Fonctionnalité du jeu contre un ordinateur	3
2.2.1	Itération 2	3
2.2.2	Itération 3	3
2.3	Fonctionnalité de personnalisation du jeu	3
2.3.1	Itération 3	3
3	Découpage de l'application en sous-systèmes	4
4	Avancement technique	5
5	Principaux choix	6
6	Problèmes rencontrés et solutions apportées	6
7	Organisation de l'équipe et mise en place des méthodes agiles	8
8	Bilans réalisés au long du projet	9
8.1	Bilan au 17/04/23	9
8.2	Bilan au 13/05/2023	9
8.3	Bilan final au 29/05/2023	9

1 Introduction

L'objectif de ce projet est de réaliser un jeu d'échec, dans lequel on pourrait affronter un autre utilisateur. Pour chaque partie, le joueur aura le choix d'affronter l'ordinateur ou un ami. De plus, nous voulons ajouter différentes fonctionnalités et possibilités de jeux pour donner envie au joueur de participer à de nouvelles parties !

Avec cette application, vous pouvez devenir un maître des échecs et améliorer vos compétences en toute sécurité et à votre rythme.

Pour réaliser ce projet, notre travail a été découpé en temps et en fonctionnalités. Pour le temps, nous avons suivi plusieurs itérations pour se fixer des objectifs à moyen terme mais aussi certaines réunions hebdomadaires pour évaluer où nous en étions dans les différents objectifs et re-répartir le travail si nécessaire. Pour le découpage du travail, nous avons essayé de travailler en binôme sur les parties les plus longues et complexes et lorsque notre travail était réalisé, nous allions aider ceux qui avaient besoin d'aide.

2 Fonctionnalités proposées et état d'avancement

2.1 Fonctionnalité du jeu principal, joueur contre joueur

2.1.1 Itération 1

Pour cette fonctionnalité nous avons dans un premier temps créé les pièces du jeu d'échecs en mettant en place les différents mouvements que chaque pièce peut faire. Chaque pièce a des caractéristiques et des déplacements particuliers, de ce fait il fallait trouver une façon optimisée de gérer ces déplacements. Pour la partie déplacement des pièces, ils ont été découpés en 3 grandes catégories : ceux qui sont linéaires, ceux qui sont diagonaux et ceux du cavalier. De plus, il a fallu créer le plateau sur lequel se déplacer. Ensuite, il fallait pouvoir lancer une partie donc nous avons créé une classe qui permet de lancer un jeu d'échec avec un ami. La partie complexe a été de mettre en place l'arbitrage pour détecter les erreurs lors d'une partie et donner les coups possibles. En effet, il a fallu penser à tous les coups que les utilisateurs peuvent imaginer et dire si cela est possible ou non. Il fallait aussi assurer la défense du roi et d'arbitrer les coups afin d'éviter les coups illégaux et de mettre en place le déplacement graphique des pièces.

Côté graphique, pour cette fonctionnalité l'idée a été de dans un premier temps de créer le plateau et de mettre des images pour chaque pièce pour pouvoir tester visuellement le code. De plus, il a fallu travailler sur le déplacement de chaque pièce en cliquant dessus puis en cliquant sur la case souhaitée. Par la suite, nous avons défini les zones autour du plateau pour visualiser les différentes parties qu'il faudra implanter. Une de ces parties était de mettre un historique des coups sur le côté de droit du plateau. Nous avons donc mis dessus quel type de pièce avait été joué, et le déplacement effectué, en notant le numéro de ligne et la lettre de la colonne.

2.1.2 Itération 2

La principale fonctionnalité de notre jeu d'échec *E7&Mat* est de pouvoir jouer sur un appareil avec un ami. Pour cela, nous avons déjà fait une première version qui comportait quelques problèmes que nous avons tenté de régler tout au long de l'itération que ça soit dans la partie jeu pur ou dans la partie graphique. Tout d'abord l'idée a été de régler les problèmes de déplacement des pièces et vérifier que tous les coups fait par les pièces sont correctes. Après une vérification théorique, nous avons testé cette implantation sur la fenêtre graphique. Pour cela nous avons lancé la partie et regardé que les coups possibles s'affichaient bien. De plus, la rotation graphique du plateau lors du changement de joueur a été mis en place. Pour réaliser toutes ces modifications, nous avons du modifier de nombreux codes et rajouter des fonctions permettant d'optimiser notre jeu.

A l'heure actuelle nous pouvons jouer une partie et elle se termine bien quand il y a "échec et mat". Cependant nous trouvons que la rotation du plateau est un peu brutale tout comme la fin de la partie qui se termine sur la fermeture instantanée en cas d'"échec et mat". Nous voulons donc améliorer cet aspect là pour le rendu final. De plus, pour tester le jeu, nous avons écrit la mise en place du jeu dans le même fichier que la partie graphique du jeu. Nous savons que cela ne respecte pas le modèle MVC (modèle/vue/contrôleur). En effet, il faudrait que nous fassions une classe seulement pour la fenêtre graphique et une autre pour le lancement du jeu en lui même. Cela demande un remaniement total de notre architecture. Ce changement est prévu mais nous ne préférons pas l'implémenter au dernier moment comme cela aurait été le cas car cela risque de créer des problèmes lors de ces changements. De ce fait, nous comptons le mettre en place au plus vite.

Côté graphique, nous avons principalement essayé de rendre un jeu graphique sans problème ce qui à parfois était un peu complexe mais nous avons actuellement une vision du jeu claire et la rotation fonctionne bien comme dit précédemment.

2.1.3 Itération 3

Lors de cette dernière itération, nous avons terminé de retirer les petits buggs que nous retrouvions pour certains coups et déplacement très précis.

Côté graphique, nous avons rajouté l'image des pièces mangées pour chaque joueur en dessous de son nom. Cette partie se trouve sous le plateau de jeu.

De plus, nous avons créé un timer. Ce dernier va tourner pour le joueur chaque fois qu'il joue et se stoppera lorsqu'il arrête de jouer.

Nous avons aussi modifier le menu initial pour qu'en lançant une partie depuis ce dernier, les joueurs rentrent leurs pseudo pour la partie. De plus, il y a la possibilité de choisir le temps que chaque joueur possède pour la partie en cours. Le choix réalisé s'appliquera aux deux joueurs.

2.2 Fonctionnalité du jeu contre un ordinateur

2.2.1 Itération 2

Lors de cette seconde itération nous avons aussi commencé à travailler sur la possibilité de jouer contre un ordinateur. Pour cela, nous avons tenté de conceptualiser le fonctionnement de cet IA, ce qui n'a pas été le plus simple. Pour le moment deux pistes sont envisagées, une IA qui fonctionne mais qui est assez lente et une autre qui est naïve. Nous avons commencé à coder cette nouvelle fonctionnalité. Pour le moment, nous sommes en phase de test donc tout ne fonctionne pas encore comme nous le souhaitons.

2.2.2 Itération 3

Durant cette dernière itération nous avons tenté de rendre fonctionnel le joueur qui n'est pas humain mais un ordinateur. Pour cela, nous avons créé un *joueurIA* qui jouerait le rôle de l'ordinateur face à qui jouer. De plus, nous avons créé des test pour voir si le joueur ordinateur répondait à nos attentes. Au final, nous n'avons malheureusement pas pu finir de régler ces soucis et le joueur n'est pas totalement opérationnel.

2.3 Fonctionnalité de personnalisation du jeu

2.3.1 Itération 3

Durant cette dernière itération, nous avons essayé de mettre en place des personnalisations pour notre jeu d'échec. Pour cela, nous avons créé en plus des pièces classiques 3 autres types de pièces. Nous retrouvons donc deux personnalisations dans l'univers de *Marvel* et une dans l'univers de *Starwars*.

De plus, nous proposons à l'utilisateur plusieurs couleurs pour personnaliser le plateau.

Nous avons aussi ajouté au jeu une musique d'ambiance comme fond sonore.

Ces trois personnalisations sont disponibles dans les réglages accessibles depuis le menu initial. Pour que tout cela fonctionne bien, nous avons dû mettre à jour le menu initial que nous avons créé précédemment pour choisir le type de partie.

Voici un bilan d'avancement des différentes fonctionnalités à la fin de ce projet. Nous pouvons constater que certaines fonctionnalités imaginées lors du premier rendu présentant ces dernières n'ont pas eu le temps d'être implantées. Cependant, les 3 fonctionnalités principales ont pu l'être ce qui est une chose dont nous sommes fiers malgré le fait que jouer contre un ordinateur ne soit pas totalement fonctionnel.

 Organisation de l'équipe	● Terminé
 Fonctionnalité : Jouer avec un ami	● Terminé
 Fonctionnalité : Jouer contre un ordinateur	● En attente
 Fonctionnalité : Mode replay	● En attente
 Fonctionnalité : Indices	● En attente
 Fonctionnalité : Personnalisation	● Planning
 Fonctionnalité : Jouer contre un ordinateur	● Planning

FIGURE 1 – Bilan d'avancement des différentes fonctionnalités

3 Découpage de l'application en sous-systèmes

Notre application se découpe en 3 grandes fonctionnalités :

- Jouer avec un ami
- Jouer contre un ordinateur (avec différents niveaux de difficulté)
- Personnalisation du jeu

Nous avons créé un unique package qui contient toutes ces fonctionnalités. En effet, nous avons créé un menu de lancement de jeu qui permet d'accéder aux différentes fonctionnalités. La grande modification entre ces 3 fonctionnalités est le type de joueur que nous affrontons qui vont tous découler de la classe "Joueur".

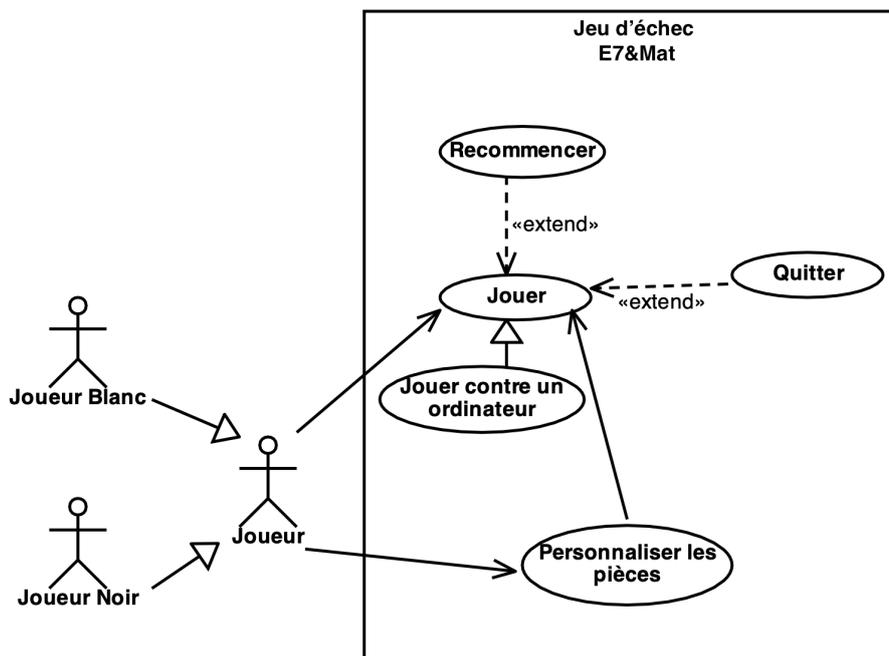


FIGURE 2 – Diagramme d'utilisation de l'application

4 Avancement technique

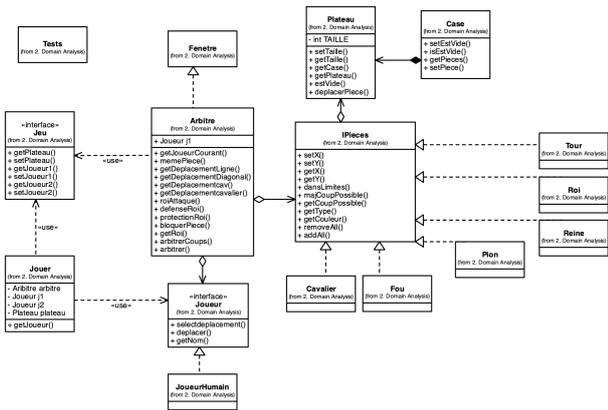


FIGURE 3 – Première version du diagramme UML

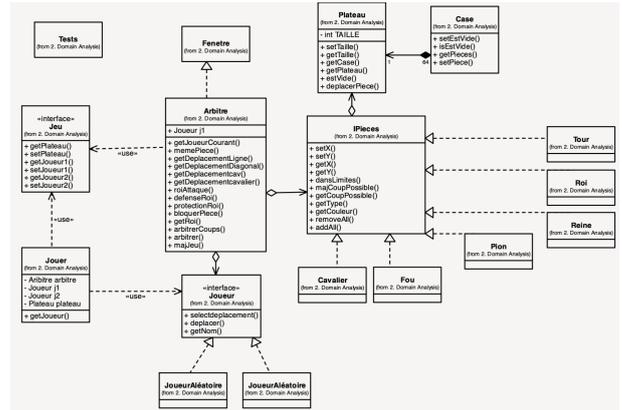


FIGURE 4 – Deuxième version du diagramme UML

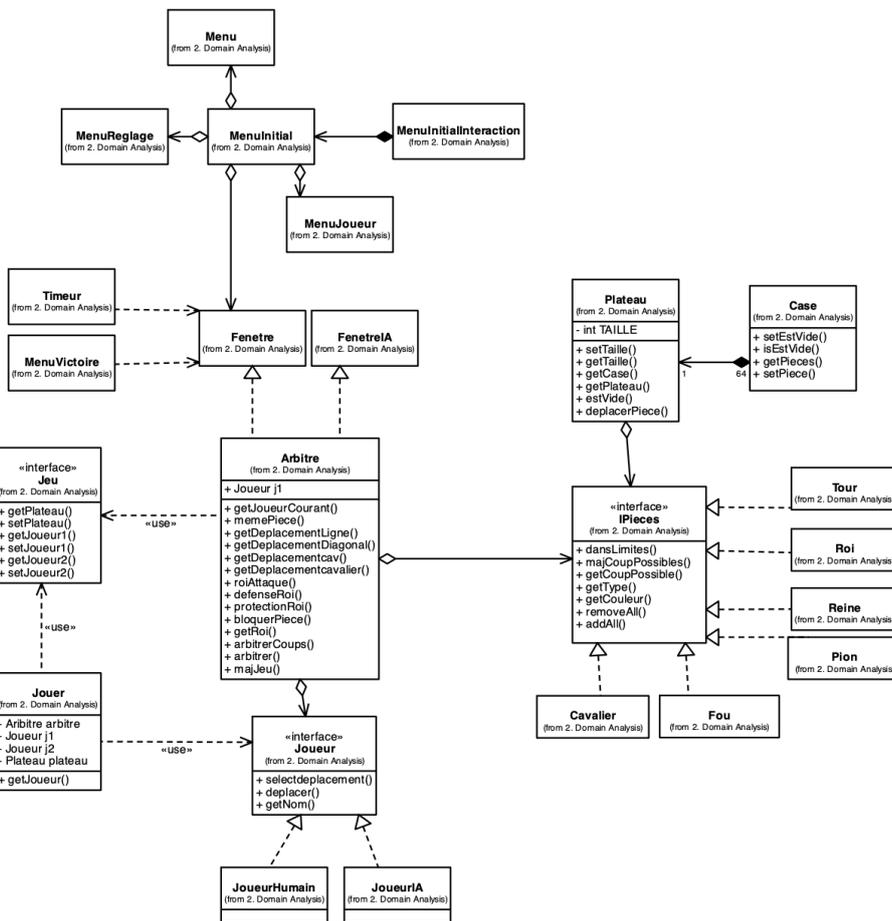


FIGURE 5 – Diagramme UML de l'application

L'implantation de l'IA repose sur l'idée que l'ordinateur ou l'IA essaie de jouer le meilleur coup possible (manger la pièce qui rapporte à son joueur le plus grand score). Il a fallu alors ajouter un nouvel attribut *valeur* pour connaître la valeur de chaque pièce. Ensuite, il fallait ajouter une fonction qui crée un arbre de décision. L'arbre de décision est créé en utilisant un type *HashMap* car nous avons eu un peu de complication à utiliser *TreeMap*. La décision du meilleur coup est basée sur une recherche sur profondeur 3. En effet, pour un état de plateau donné, nous choisissons les 4 meilleurs coups

parmi tous les coups qui sont autorisés. Ensuite, nous imaginons que l'adversaire joue aussi. Nous choisissons de prendre 3 de ses meilleurs coups pour lesquels il faudra répondre cette fois avec le meilleur coup de l'IA. Pour pouvoir faire ça, nous avons dû ajouter une fonction *getPiecesPresentes* qui retourne les pièces précédentes associées au joueur courant.

Nous avons mis en place un plateau qui contient des cases qui contiennent elles mêmes différentes données telles que les coordonnées, le type, la couleur et la valeur d'une pièce. Nous avons également mis en place les différentes pièces avec des fonctions qui leurs sont propres, leur permettant d'avoir les bons déplacements et les caractéristiques qui leurs sont inhérentes. Nous avons également mis en place un arbitrage en plusieurs étapes qui permet d'affiner la légalité des coups possibles, assurer une bonne défense du Roi (à travers le déplacement du Roi si il est attaqué ou l'interception du attaque en mettant une pièce protégeant le roi).

5 Principaux choix

Dans un premier temps nous avons décidé que les pièces d'échecs seraient une interface et que chaque pièce implémenterait cette interface comme nous pouvons le voir sur le digramme UML (figure 5).

Pour les déplacements, il a été choisi de créer 3 méthodes de déplacements pour les pièces : les linéaires (horizontaux et verticaux), les diagonaux et ceux du cavalier (qui ont un mouvement en "L").

Pour la partie graphique, nous avons décidé de mettre la grille au centre de la fenêtre graphique et de montrer chaque coups réalisés au fur et à mesure de la partie. Nous avons aussi fait le choix de créer un menu initial au jeu pour accéder aux différentes fonctionnalités.

Pour le plateau, nous l'avons considéré comme un plateau de coordonnées.

De même, nous avons codé les coups possibles comme une collection contenant ces coups.

Dans le cas des coups rendus légaux nous avons décidé de traiter indépendamment la protection du roi, le déplacement du roi et celui des autres pièces.

Nous avons également décidé de traiter l'ensemble du plateau à chaque tour, au lieu de traiter uniquement une partie.

Nous avons également choisi de traiter les coups illégaux avec plusieurs vagues, qui les supprimait petit à petit.

6 Problèmes rencontrés et solutions apportées

Le principal problème rencontré a été le MVC (Modèle/View/Contrôleur). En effet, lorsque nous avons commencé la mise en place du projet nous n'avions pas bien saisi comment le mettre en place de ce fait il a fallu tout reprendre car sinon nous rencontrions des problèmes lors du lancement du jeu depuis le menu.

Un problème fut aussi de bien comprendre les subtilités d'un jeu d'échec pour pouvoir le coder et bien comprendre les coups et techniques classiques connues.

Un autre problème fut le déplacement des pièces. En effet, nous nous sommes rendu compte grâce à des tests unitaires que certaines pièces ne se déplaçaient pas exactement comme nous le souhaitions. Nous avons aussi réalisé des tests d'intégrations pour essayer de bien comprendre ce qu'il se passait et ensuite, il a fallu bien relire tout ce qui avait été fait pour trouver les endroits du code qui posaient problème.

Pour le plateau, nous avons dû créer plusieurs fonctions qui nous permettent d'accéder aux informations du plateau car notre plateau est composé de cases et dans chaque cas nous retrouvons une pièce ou non.

Nous avons fait face à de nombreux problèmes lors de l'arbitrage de la partie. En effet, le roi pouvait être mis en échec à la découverte d'une pièce. Pour pallier à ce problème, il a fallu voir un coup en avance dans le futur afin de s'assurer qu'il ne soit pas vulnérable si les pièces autour de lui venait à bouger.

Pour la partie graphique, nous avons eu quelques problèmes au début pour créer ce que nous souhaitions car il a fallu bien comprendre le fonctionnement de Swing. Pour cela, nous avons travaillé avec le cours et beaucoup de recherches sur

la documentation pour essayer d'avoir un rendu intéressant.

Nous avons également eu du mal à gérer l'affichage des éléments sur différents écrans. En effet, il fallait que le plateau soit carré pour ne pas avoir de problèmes d'affichages au niveau des cercles qui montrent les coups possibles pour une pièce. Pour résoudre ce problème, nous avons ajouté un listener à la fenêtre pour qu'à chaque fois que nous modifions la taille de la fenêtre principale, nous fixions la taille du panel contenant le plateau et que nous calculions les tailles des panels au nord, sud, est et ouest en tenant compte de cette contrainte.

De même, nous avons dû adapter la taille des icônes des boutons pour le menu de réglage afin qu'ils s'affichent correctement sur tous les écrans. Nous avons donc ajouté une méthode qui permet de redimensionner une icône directement dans le code du menu pour ne pas avoir à dimensionner les icônes à la main.

Une difficulté rencontrée était de récupérer au sein de la fenêtre globale de la partie les informations enregistrées dans les menus concernant les réglages, le nom des joueurs et le temps des parties. Pour ce faire, nous avons utilisé des variables "static" pour qu'elles soient propre à la classe des menus. Ainsi, pour récupérer les variables, nous avons créé une nouvelle instance des menus dans la fenêtre en passant par un autre constructeur pour pouvoir obtenir le contenu des variables enregistrées dans une autre instance de la classe des menus.

Nous avons également rencontré des problèmes lors du traitement des coups légaux car nous devions considérer une profondeur plus basse. Cependant cela modifiait le jeu pendant les calculs ce qui nous renvoyait des erreurs. Nous avons décidé de traiter ces coups en plusieurs temps, ce qui nous a permis de descendre en profondeur par palier et éviter les erreurs précédemment rencontrées.

Nous avons aussi rencontré à la fin une difficulté pour implémenter l'IA car à défaut de pouvoir utiliser TreeMap on a du utiliser un type HashMap pour tous les meilleurs coups sauvegardés ce qui a fait que le type faisait 1 ligne complète à écrire à chaque fois.

7 Organisation de l'équipe et mise en place des méthodes agiles

Notre équipe s'est organisée au fur et à mesure selon les besoins et problèmes rencontrés. Nous avons découpé notre application en différentes fonctionnalités et ces dernières ont été découpées en épics. Chaque membre du groupe travaillait sur un épique. Nous avons travaillé principalement en binôme, cependant il est arrivé que certains épics soient réalisés par une unique personne. De plus, nous avons planifié des réunions hebdomadaires pour discuter de l'avancement de chacun et pouvoir rajouter des personnes pour travailler sur certaines parties plus complexes. De plus, nous avons pu ainsi débattre sur certains points techniques de la façon dont nous allions les traiter.

Nous avons tout d'abord travaillé sur la partie jouer aux échecs avec un ami sur un même écran. Pour cela, nous avons découpés cette fonctionnalité en différents épics que nous nous sommes répartis.

Par la suite, nous avons avancé sur les deux autres fonctionnalités majeures soit la personnalisation et la possibilité de jouer contre un ordinateur. Pour cela, nous avons procédé de la même manière.

Pour bien mettre en place ces méthodes agiles, nous avons utilisé principalement deux outils : discord et notion.

Discord a été notre outil de communication tout au long du projet. Nous avons ainsi pu faire nos réunions en visio ou en vocal dessus chaque semaine lorsque nous ne pouvions pas nous voir en physique. De plus, il permet de travailler à plusieurs en faisant un partage d'écran. Nous avons aussi créé des "channels" de discussion spécifique à certaines parties quand nous avions besoin de ne pas perdre certaines informations.

Notion a été notre outil pour nous organiser. Nous avons ainsi noté nos réunions dessus et l'organisation de l'équipe en général. De plus, ce site nous permet de créer un événement ou une fonctionnalité et de la découper en plusieurs tâches à réaliser, d'y affecter des personnes mais aussi une date d'échéance. Nous avons aussi listé les éléments clés de la fonctionnalité. Ce fut un outil très pratique pour mettre les différents éléments dont nous avions besoin et cela nous a permis de mettre à jour ce que nous faisons. En effet, chaque tâche est indiquée comme "en cours", "terminé" ou "pas commencé" à l'aide d'un statut de la tâche. Voici un exemple qui présente le fonctionnement de cet outil :

The screenshot shows a Notion table with the following columns: 'Nom de la tâche', 'Status', 'Attribution', 'Échéance', and 'Projet'. The table lists 14 tasks, most of which are 'Terminé' (Completed) and one is 'Pas commencé' (Not started). The tasks include weekly meetings and milestones for iterations. The attribution column lists team members like H. Yassine, Ines Charles, Nouha Lahyen, and Alphonse Lafon. The deadline column shows dates from 17 mars 2023 to 29 mars 2023. The project column is labeled 'Organisation de l'équipe'. At the bottom, there is a progress indicator: 'TERMINE 92,857 %'.

Nom de la tâche	Status	Attribution	Échéance	Projet
Réunion de répartition du travail	Terminé	H. Yassine, Ines Charles	17 mars 2023	Organisation de l'équipe
Mise en place du Notion	Terminé	Ines Charles, H. Yassine	17 mars 2023	Organisation de l'équipe
Mise en place du git	Terminé	Nouha Lahyen, Ines Charles	17 mars 2023	Organisation de l'équipe
Réunion de mise au point	Terminé	Alphonse Lafon, Ines Charles	26 mars 2023	Organisation de l'équipe
Réunion de mise au point	Terminé	Alphonse Lafon, Ines Charles	2 avril 2023	Organisation de l'équipe
Réunion de mise au point	Terminé	Alphonse Lafon, Ines Charles	9 avril 2023	Organisation de l'équipe
Réunion de fin de 1ère itération	Terminé	Alphonse Lafon, Ines Charles	16 avril 2023	Organisation de l'équipe
Réunion de mise au point	Terminé	Alphonse Lafon, Ines Charles	23 avril 2023	Organisation de l'équipe
Réunion de mise au point	Terminé	Alphonse Lafon, Ines Charles	30 avril 2023	Organisation de l'équipe
Réunion de mise au point	Terminé	Alphonse Lafon, Ines Charles	7 mai 2023	Organisation de l'équipe
Réunion de fin de 2ème itération	Terminé	Alphonse Lafon, Ines Charles	14 mai 2023	Organisation de l'équipe
Réunion de mise au point	Terminé	Alphonse Lafon, Ines Charles	21 mai 2023	Organisation de l'équipe
Réunion de fin de 3ème itération	Terminé	Alphonse Lafon, Ines Charles	27 mai 2023	Organisation de l'équipe
Réunion préparation à l'oral	Pas commencé	Alphonse Lafon, Ines Charles	29 mars 2023	Organisation de l'équipe

FIGURE 6 – Exemple de présentation de l'outil notion : notre organisation divisée en différentes tâches

17 Organisation de l'équipe

État **En cours**

Personnes Vide

Achèvement 92,9 %

Dates 15 mars 2023 → 18 mars 2023

1 propriété de plus

Ajouter un commentaire...

About this project

- Mise en place des outils
- Dates des réunions

Project tasks

FIGURE 7 – Visualisation de cette organisation

+ Project tasks

Aa Nom de la tâche	Attribution	Projet
Mise en place du Notion	<ul style="list-style-type: none">Ines CharlesH. Yassine romainNouha LahyenAlphonse Lafonyanis GUIRENE	Organisation de l'équipe
Mise en place du git	<ul style="list-style-type: none">Nouha LahyenInes CharlesH. Yassine romainAlphonse Lafonyanis GUIRENE	Organisation de l'équipe
Réunion de répartition du travail	<ul style="list-style-type: none">H. YassineInes CharlesromainNouha LahyenAlphonse Lafonyanis GUIRENE	Organisation de l'équipe

FIGURE 8 – Visualisation de cette organisation suite

8 Bilans réalisés au long du projet

8.1 Bilan au 17/04/23

Actuellement, nous avons une première version d'échecs pour jouer entre amis. Il reste encore certains détails à régler pour qu'il fonctionne parfaitement. De plus, nous allons maintenant essayer de mettre en place les bots pour pouvoir jouer contre eux. Pour le côté graphique, nous allons créer ce qu'il faut mettre dans les espaces déjà délimités (le timer, le bilan, ect...). Une autre idée est de mettre en place la rotation du plateau quand la personne qui joue change. De plus, l'objectif est de créer un menu de début d'application qui nous permettra d'accéder aux différents modes de jeu possibles.

8.2 Bilan au 13/05/2023

Nous avons une version fonctionnelle du jeu principal. De plus, nous avons commencé à mettre en place une version pour jouer contre une intelligence artificielle. Nous avons créé un menu initial pour le jeu qui nous permettra à terme de lancer les différents modes de jeu que nous proposons et d'accéder aux réglages.

D'ici la fin du projet, nous avons comme objectif de remanier l'architecture pour qu'elle soit plus cohérente et fonctionne mieux. De plus, nous voulons finir le contour de notre fenêtre de jeu et rajouter la possibilité de jouer contre les IA que nous serons, nous l'espérons, opérationnels d'ici là. Enfin, nous voulons mettre en place la partie réglage qui permettra de personnaliser notre jeu d'échecs. A l'heure actuelle, nous avons commencé à créer 2 nouveaux types de pièces que nous pensons rajouter au jeu d'ici la version finale.

8.3 Bilan final au 29/05/2023

Nous avons une application qui se lance sur un menu initial qui nous permet d'accéder à nos trois principales fonctionnalités. De plus, nous avons un jeu d'échec pour jouer avec un ami sur un même appareil qui fonctionne totalement et qui est personnalisable. Nous pouvons en effet choisir la couleur du plateau, les pièces utilisées et la durée du jeu. Nous avons aussi mis en place un fond sonore qu'il est possible d'activer et de désactiver dans les réglages.

La mise en place du jeu contre un ordinateur est en cours. En effet, le joueur IA qui est notre ordinateur est codé et peut jouer en réaction aux actions du joueur mais il n'est pas encore très performant.

Nous avons aussi pris le temps de remanier le code afin qu'il soit plus conforme aux règles de codage étudiées en cours tout au long du semestre.