

Rapport CYPATH

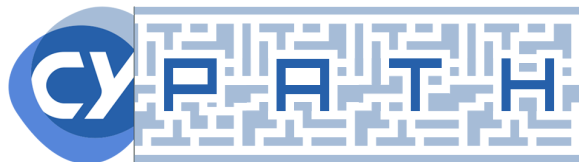
Chantôme Amandine

Fourcher Line

Guillerm François

Sing Valentin

May 2023



Contents

1	Introduction	3
2	Organisation de l'équipe	4
2.1	Formation du groupe	4
2.2	Outils utilisés	4
2.3	Répartition des tâches	4
2.4	Méthodes et rythme de travail	4
3	Problématiques rencontrées	5
3.1	Représentation du plateau de jeu	5
3.2	Algorithme de recherche	5
3.3	Vérification des mouvements	5
3.4	Transition vers l'interface graphique	5
3.5	Outils de développement	5
3.6	Git	5
4	Limitations et améliorations	7
4.1	Ergonomie	7
4.2	Sauvegardes	7
4.3	Jouer tout seul	7
4.4	Affichage de la fenêtre	7
5	Planning du projet	8
6	Diagrammes	9
6.1	Diagramme de cas d'utilisation	9
6.2	Diagramme de classe	10
6.3	Diagramme de séquence	11

1 Introduction

Le projet que nous avons choisi est CYPATH. Le but de ce projet est de reproduire le jeu du Quoridor en Java, avec une interface graphique. Dans ce jeu, il faut atteindre l'autre côté du plateau avec son pion, avant son ou ses adversaires. Pour ce faire, le joueur peut soit déplacer son pion d'une case, soit placer une barrière pour ralentir son adversaire, ou faciliter son chemin.

2 Organisation de l'équipe

2.1 Formation du groupe

Pour ce projet, nous avons formé un groupe de 4. Nous nous connaissions déjà tous et savions les forces et les faiblesses de chacun. Nous avons déjà travaillé ensemble sur de précédents projets, cela nous semblait donc être une bonne idée de former ce groupe pour ce projet.

2.2 Outils utilisés

Nous avons utilisé plusieurs outils pour ce projet :

- Pour communiquer entre nous sur l'avancement du projet, nous avons utilisé Discord. Cela nous a notamment permis d'organiser des réunions régulières.
- Pour les réunions organisées avec notre tuteur de projet, nous avons Teams.
- Pour les diagrammes UML, nous avons utilisé diagrams.net.
- Pour réaliser une maquette des différentes scènes du projet, nous avons utilisé Figma.
- Pour programmer, nous avons utilisé l'IDE IntelliJ.
- Et enfin, pour la gestion de version de notre code, nous avons utilisé Github.

2.3 Répartition des tâches

Nous n'avions pas de répartition claire tout au long du projet. La plupart du temps, lorsque quelqu'un finissait quelque chose, il ou elle en commençait une autre en fonction de ce qu'il restait à faire, même si cela n'avait pas vraiment de rapport avec ce qu'il ou elle venait de faire. Il y a tout de même certains éléments qui ont été fait en grande partie par certains membres en particulier :

- Amandine a travaillé sur la maquette de l'interface graphique, sur la sauvegarde et le chargement de parties, sur la sérialisation et en partie sur l'interface graphique en elle-même.
- François a travaillé sur le graphe et donc sur les classes position et edge, sur la sauvegarde et le chargement de parties, sur les scènes du menu, et en partie sur l'interface graphique.
- Line a travaillé sur la maquette de l'interface graphique et en grande partie sur l'interface graphique.
- Valentin a travaillé sur la classe player, notamment en faisant tous les tests nécessaires pour savoir si un joueur peut déplacer son pion ou placer une barrière à un certain endroit et en très grande partie sur l'interface graphique, notamment sur l'affichage des mouvements possibles pour les joueurs et du placement des barrières.

2.4 Méthodes et rythme de travail

Nous avons travaillé pratiquement tous les jours durant ces deux semaines de projets. La plupart des jours, nous ne commençons pas à travailler avant 14 heures environ, et nous finissons de travailler vers 23 heures, voire plus tard. Durant ces périodes de travail, nous prenions bien évidemment des pauses.

Nos méthodes de travail sont assez basiques. La plupart du temps, nous travaillions par groupe de deux, et à chaque fois que nous finissions quelque chose, nous le mettions sur Git. Nous avons aussi travaillé en groupe complet à plusieurs reprises, que ce soit pour des parties du programme sur lesquelles nous voulions réfléchir à plusieurs, ou tout simplement pour être plus efficace.

3 Problématiques rencontrées

3.1 Représentation du plateau de jeu

Tout au long du projet, nous avons été confrontés à plusieurs problématiques qui ont nécessité des efforts supplémentaires pour les résoudre. La première problématique était liée à la représentation du plateau de jeu. Nous étions confrontés à un choix entre l'utilisation d'une matrice ou d'un graphe pour représenter les cases et leur connexions. Nous avons finalement opté pour un graphe pondéré où le poids de chaque liaison changeait au placement d'une barrière.

3.2 Algorithme de recherche

Une autre problématique importante était liée à l'algorithme de recherche à utiliser pour connaître l'existence d'un chemin possible vers l'arrivée. Nous ne savions pas s'il fallait utiliser un algorithme de recherche plus robuste ou un simple parcours en profondeur (DFS). Nous avons également la possibilité d'utiliser un algorithme de recherche du plus court chemin comme Dijkstra. Comme nous voulions seulement savoir si un chemin existait entre le joueur et l'arrivée nous avons décidé d'utiliser l'algorithme le plus simple c'est-à-dire le DFS.

3.3 Vérification des mouvements

La vérification des mouvements des pions a également posé problème. Nous devons prendre en compte des contraintes spécifiques, telles que les mouvements en diagonale ou les sauts de pions. La mise en place de ces vérifications s'est révélée plus complexe que prévu, nous avons finalement vérifié chaque mouvement possible du pion individuellement, si le mouvement était possible alors il est valide.

3.4 Transition vers l'interface graphique

Une autre difficulté rencontrée était le passage de la version non graphique du jeu à une version graphique. Nous avons sous-estimé la complexité de cette tâche et avons dû consacrer plus de temps que prévu pour mettre en place l'interface graphique voulue.

3.5 Outils de développement

En outre, nous avons rencontré des problèmes techniques spécifiques à nos outils de développement. Certains membres de l'équipe ont fait face à des difficultés avec IntelliJ et ont dû investir du temps supplémentaire pour résoudre ces problèmes. De plus l'apprentissage de l'outil Scene Builder pour la création de l'interface graphique a également nécessité une prise en main et une gestion des problèmes techniques rencontrés lors de son utilisation engendrant une perte de temps conséquente. Nous ne pouvions pas l'utiliser à plusieurs simultanément en session "Code with me" sur IntelliJ, ce qui a compromis l'efficacité du travail de groupe.

3.6 Git

Enfin, l'intégration de Git dans notre flux de travail a été une autre problématique à surmonter. Certains membres de l'équipe, en particulier François, ont dû se familiariser avec les fonctionnalités et les bonnes pratiques de Git, ce qui a demandé un investissement en temps et en apprentissage supplémentaire.

Malgré ces problématiques, nous avons réussi à les surmonter grâce à une collaboration étroite, à une communication ouverte et à des efforts d'apprentissage continus. Ces difficultés nous ont permis de grandir en tant qu'équipe et d'acquérir de nouvelles compétences tout au long du projet.

4 Limitations et améliorations

4.1 Ergonomie

Le projet présente certaines limitations fonctionnelles et de performance qui pourraient être améliorées. Sur le plan ergonomique et fonctionnel, il y a quelques aspects à prendre en considération. Par exemple, lorsqu'un joueur est en cours de partie et souhaite retourner à l'écran d'accueil, il serait judicieux de demander une confirmation pour éviter les actions accidentelles et perdre la partie en cours sans le vouloir.

Une autre amélioration serait de permettre au joueur de personnaliser certains éléments de la partie. Actuellement, lorsqu'un joueur lance une partie, il n'a pas la possibilité de choisir son pion, sa couleur ou la taille du plateau. En ajoutant cette fonctionnalité, les joueurs pourraient avoir une expérience plus personnalisée et adaptée à leurs préférences.

4.2 Sauvegardes

En ce qui concerne les sauvegardes, il serait utile d'afficher plus d'informations sur les parties sauvegardées. Par exemple, inclure le nombre de joueurs, la durée de la partie, ou d'autres détails pertinents, afin que les joueurs puissent facilement identifier la sauvegarde souhaitée.

4.3 Jouer tout seul

Une évolution intéressante serait d'offrir la possibilité à un joueur de jouer contre une IA en tant que joueur unique, avec la possibilité de sélectionner le nombre d'adversaires virtuels (2, 3 ou 4 joueurs). Cela permettrait aux joueurs de profiter du jeu même en l'absence de partenaires réels.

4.4 Affichage de la fenêtre

La gestion de la fenêtre d'affichage constitue également un aspect important à améliorer. Actuellement, l'application est conçue pour un écran de résolution fixe (1920 x 1080). Pour une meilleure expérience utilisateur, il serait préférable d'adopter une approche plus dynamique qui s'adapte à différentes tailles d'écran. Cela permettrait aux utilisateurs de profiter pleinement du jeu, quelles que soient les dimensions de leur écran.

5 Planning du projet

Lundi 15	- Création des diagrammes UML, mise en place du projet avec Git
Mardi 16	- Création du graphe et des classes associées - Maquettage des scènes de l'interface graphique
Mercredi 17	- Création du graphe et des classes associées - Maquettage des scènes de l'interface graphique - Création de la classe Player, vérification du déplacement d'un joueur
Jeudi 18	- Création du graphe et des classes associées - Création de la classe Player, vérification du déplacement d'un joueur
Vendredi 19	- Création de la classe Player, vérification du déplacement d'un joueur - Création de l'affichage du plateau de jeu
Samedi 20	- Création de l'affichage du plateau de jeu - Création des méthodes permettant de sérialiser et désérialiser une partie
Dimanche 21	
Lundi 22	- Affichage partie en cours et menu
Mardi 23	- Affichage partie en cours et menu - Sauvegarde et chargement d'une partie
Mercredi 24	- Finalisation de l'interface graphique - Sauvegarde et chargement d'une partie
Jeudi 25	- Tests et correction de bugs
Vendredi 26	- Tests et correction de bugs
Samedi 27	- Rédaction du README - Vérification des commentaires
Dimanche 28	- Finalisation du README - Rédaction du rapport final

Figure 1: Planning du projet

6 Diagrammes

6.1 Diagramme de cas d'utilisation

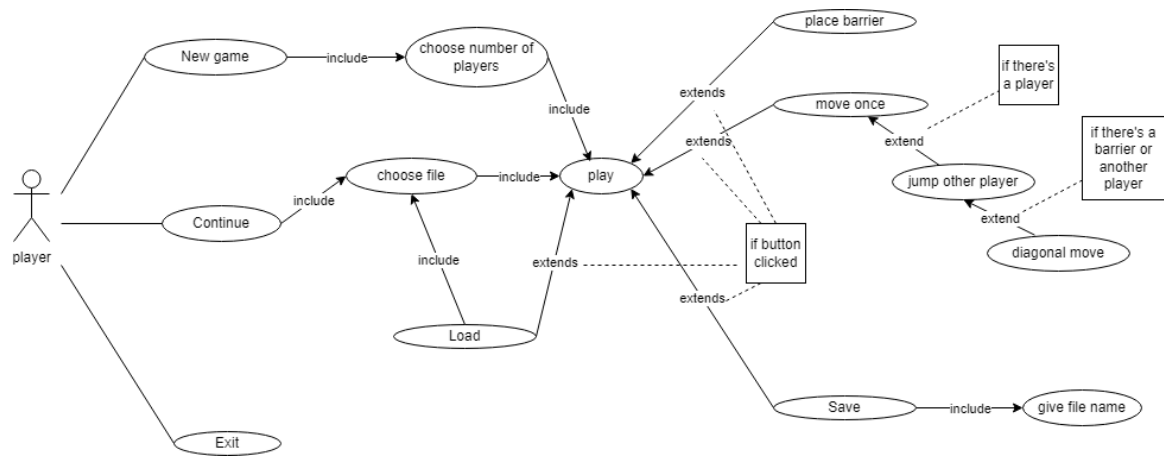


Figure 2: Diagramme de cas d'utilisation

6.2 Diagramme de classe

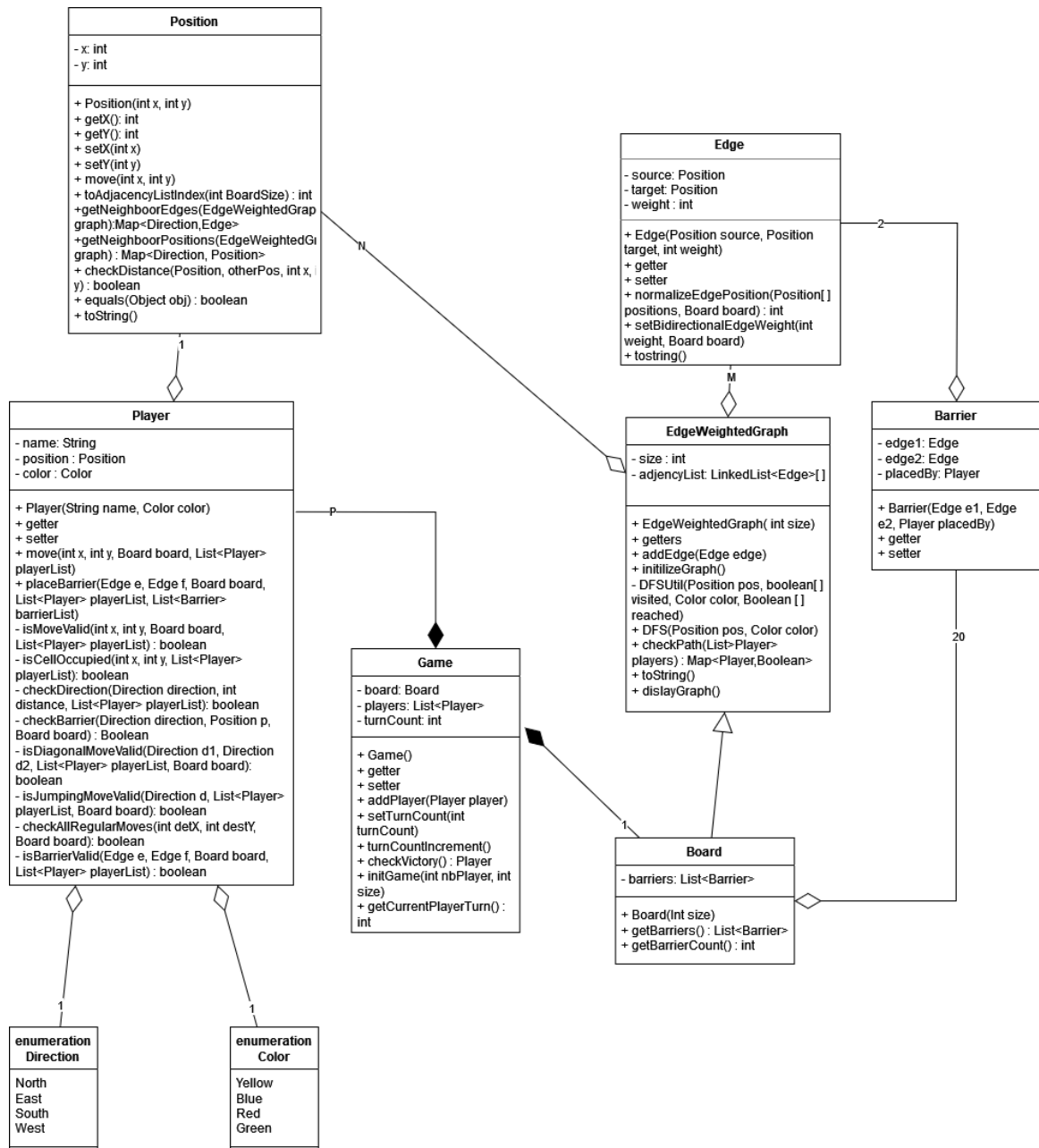


Figure 3: Diagramme de classe

6.3 Diagramme de séquence

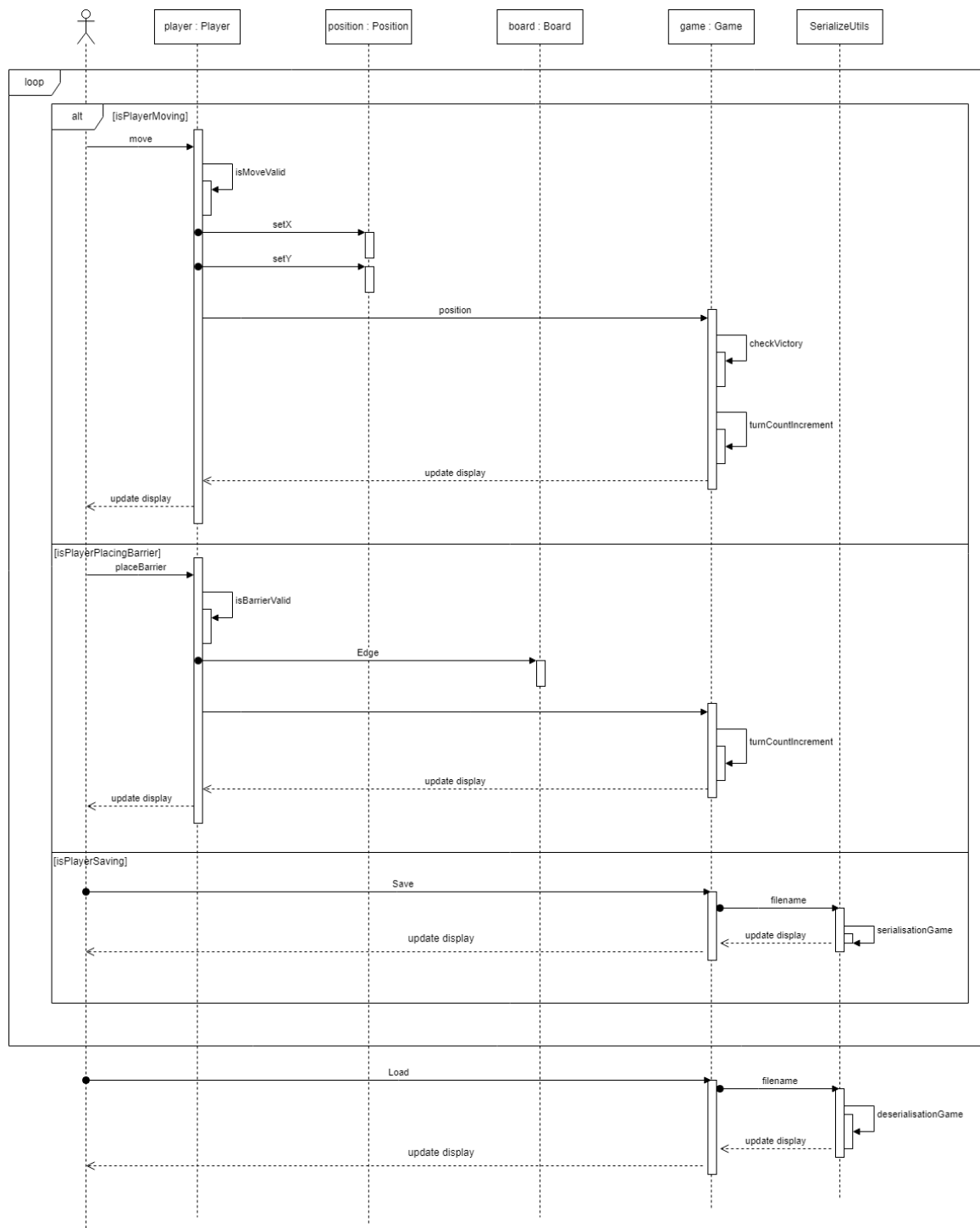


Figure 4: Diagramme de séquence